2-11-1994

Laser Resonators Using Tiered Fresnel Mirrors

Bruce Dale Ulrich
Portland State University

THESIS APPROVAL

The abstract and thesis of Bruce Dale Ulrich for the Master of Science in Electrical Engineering were presented February 11, 1994 and accepted by the thesis committee and the department.

COMMITTEE APPROVALS:

[redacted]
Lee Casperson, Chair

[redacted]
Branimir Pejcinovic

[redacted]
Carl Bachhuber
Representative of the Office of Graduate Studies

DEPARTMENT APPROVAL:

[redacted]
Rolf Schaumann, Chair
Department of Electrical Engineering

* *

ACCEPTED FOR PORTLAND STATE UNIVERSITY BY THE LIBRARY

by [redacted] on 22 March 1994

ABSTRACT

An abstract of the thesis of Bruce Dale Ulrich for the Master of Science in Electrical Engineering presented February 11, 1994.

Title: Laser Resonators Using Tiered Fresnel Mirrors.

A reflective Tiered Fresnel Zone Plate, herein called a Tiered Fresnel Mirror TFM, with a focal length on the order of a meter is studied for use as the mirror(s) in a Fabry-Perot interferometer type of laser. The relative phase transition within the individual zones (ideally smooth from zero to pi) is stair-stepped or tiered in the longitudinal direction of the mirror. Within an individual zone the step height is constrained to a constant whereas the width of the tiers are monotonically decreased when traversing radially outward so that the overall profile follows the ideal smooth curve. The effectiveness of the number of tiers per zone, measured by the loss per pass or round-trip, varies from a Plane Mirror (zero tiers per zone) to a Spherical Mirror (an infinite number of zones per tier).

The Fox and Li iterative method of determining the E-Field as the beam propagates back and forth is applied to an empty cavity resonator to determine the diffraction loss. A computer program is written to investigate the diffraction loss of various mirror configurations. The performance of the TFM is found to be not as efficient as the Spherical Mirror (the number of tiers per zone is shown to be a major variable) but may be tolerable under applications of a moderately high gain laser

medium. The Gaussian Fundamental mode is easier to maintain since the higher order modes have a higher loss per round trip.

The manufacture of the TFM can be incorporated easily into an IC process thereby making the cost of the novel mirror relatively cheap when produced in quantities. A major cost variable is again the number of tiers per zone which is proportional to the number of processing steps. The TFM's performance with respect to the etch depth of the steps in the mirror's stair-stepped profile is simulated and found to be a very doable etch with the current plasma etch technology.

LASER RESONATORS

USING TIERED FRESNEL MIRRORS

by

BRUCE DALE ULRICH

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
ELECTRICAL ENGINEERING

Portland State University
1994

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The Fresnel Zone Plate is a planar optical element with lens-like properties that can focus electromagnetic waves[1]. It is a transmission device, the electromagnetic waves transmit through it, focusing by diffraction and interference rather than by refraction.

There are various types of Fresnel Zone Plates as illustrated in Figure 1. In section A of the figure an amplitude type is shown where each alternate concentric ring is fully opaque (or non-transmitting) while the other rings are fully transmitting. The efficiency is not very good since close to 50% of the light is lost at the onset in front of the zone plate. In section B the Planar Lens is shown with two (or more) dielectrics being used in alternate concentric rings. The ideal is to have a relative phase difference of one-half of a wavelength between adjacent rings. This is accomplished by the different dielectric constants of the materials. A proper choice of parameters results in a planar zone plate of constant thickness. In section C is a Phase-reversing Zone Plate. This is essentially the same concept as the Planar Lens Zone Plate of section B yet planarity is of no concern. In section D a Quarter-period Zone Plate is shown. Here each concentric ring is phase corrected in a stair-stepped or tiered manner. The correction spans from zero to pi radians of an optical cycle, that is a wavelength, and the zero is at the inner radius and the pi is at the outer radius of each ring. This particular zone plate is stair-stepped with four tiers per ring, thus it is called a Quarter-period Zone Plate. In section E a Fresnel Lens is shown which is similar to the Quarter-period Zone Plate yet it has an infinite number of tiers within each ring making the profile a smooth transition from the inner to the outer part of each ring. This makes the phase perfectly corrected across each ring. The efficiency is nearly equal to the Simple Lens as is shown in section F of Figure 1.

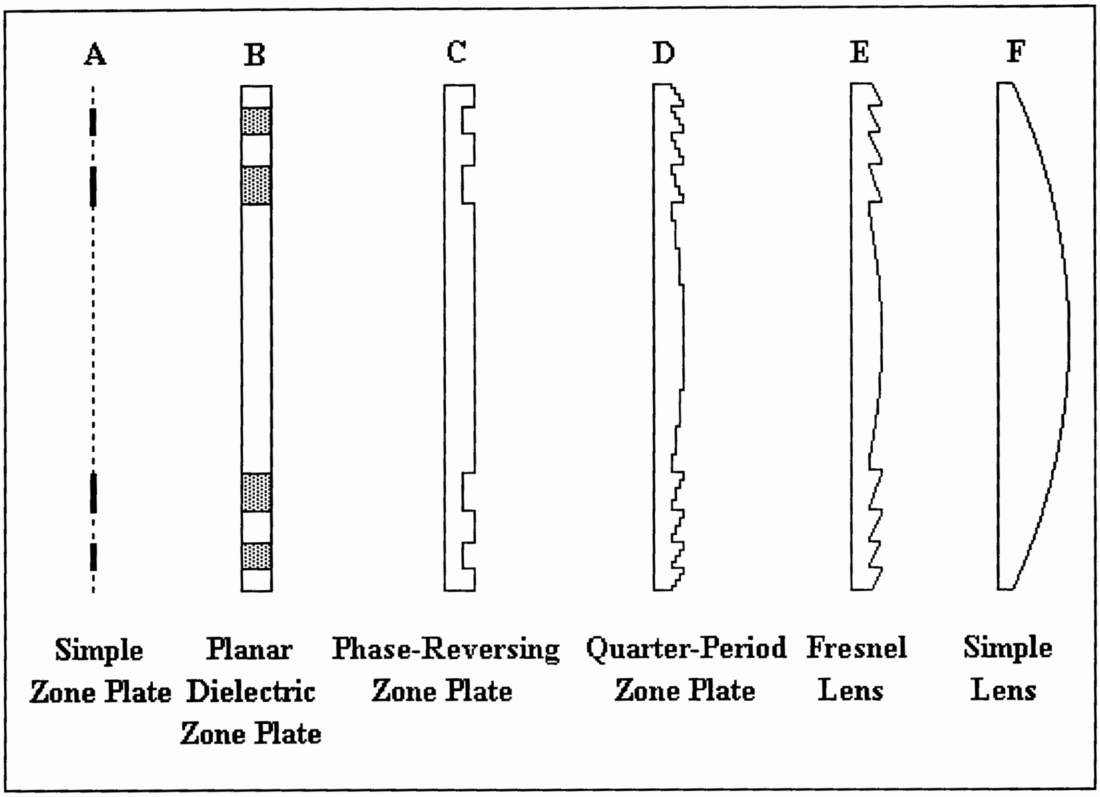


Figure 1. Various types of Zone Plates and Lenses.

Phase-correcting Fresnel Zone Plates are of particular interest since this principle is applied to a reflective type of zone plate described herein[2]. Micro-Fresnel Lenses with small apertures and large numerical apertures are required components in various optical systems such as pickup lenses in laser-disk players and coupler lenses in optical elements of optical communication systems. These have been made by laser and electron beam lithography[3,4]. A Fresnel Lens has been produced in an Integrated Circuits process on an oxidized silicon substrate using a silicon nitride waveguide[5]. A good review of reflection zone plates can be found in an article by Garrett and Wiltse[6].

Aspherical mirrors that are used in an optical resonator have been analyzed such that a large fundamental-mode beam width might be produced with a large transverse-mode discrimination[7].

The heart of this thesis is the study of a novel mirror called a Tiered Fresnel Mirror. It is used as the mirror component in an Optical Resonator. Generally, the mirrors at each end in a common laser system, that being basically a spherical mirror Fabry-Perot interferometer, are either both Spherical Mirrors or a combination of a Spherical and a Plane Mirror. Replacement of the Spherical Mirror with the Tiered Fresnel Mirror will be shown to be viable. Thus comparison of the Spherical Mirror to the Tiered Fresnel Mirror will be of utmost importance.

There already is the term "Fresnel Mirrors" used to describe two Plane Mirrors inclined at an angle to each other that produce an interference pattern [8]. No reference or use of these "Fresnel Mirrors" are contained in this thesis (only the terms "Fresnel Mirror" and "Tiered Fresnel Mirror" are used to identify another type of mirror). The Fresnel Mirror and the Tiered Fresnel Mirror will be defined in Chapter III and use of these terms will pertain to the definition given therein.

We will begin with a background discussion of the theory of Optical Resonators in Chapter II. This is a backbone chapter required for any study on Optical Resonators. Ray-Transfer Matrices are developed and the Stable Spherical Resonator is defined. Some important Symmetric Mirror Resonators are discussed. Finally, TEM Resonator Modes are defined.

Chapter III defines the Spherical Mirror, the Fresnel Mirror, and the Tiered Fresnel Mirror. The Fresnel Number N, an important number involved with the mirror loss due to diffraction, is discussed. Diffraction loss is also discussed and is the Figure of Merit for the performance of any mirror. Finally, the Fox and Li Method is used to determine the loss per pass in a Symmetric Resonator or the loss per round-trip in a Non-

Symmetric Resonator. This method uses the Huygens-Fresnel Diffraction Integral via an iterated means to an end, namely the Diffraction Loss.

A computer program was written to perform the Fox and Li Method for many different resonator configurations. These computer simulations are contained in Chapter IV. In this chapter the comparison of the performance is made between the Spherical Mirror and : 1) the Fresnel Mirror; and 2) the Tiered Fresnel Mirror.

Integrated Circuits Processing is the key to the afford ability of the Tiered Fresnel Mirror. Chapter V is devoted to the discussion of incorporating the Tiered Fresnel Mirror instead of the Solid State Chip into an Integrated Circuits Process along with its benefits.

Chapter V is the conclusion chapter. The novel Tiered Fresnel Mirror will be deemed viable and affordable when manufactured with an Integrated Circuits Process.

A listing of the computer program "RESONATE ver 1.0" in included in Appendix A along with a cross-reference map of all the variables used in Appendix B for those interested in further work involved in diffraction integrals and/or the Fox and Li Method.

Finally, an appreciation goes to Dr. Lee Casperson for some helpful discussions, and his kindness and patience was warmly appreciated.

CHAPTER II

OPTICAL RESONATOR THEORY

MATRIX OPTICS

Ray optics or geometrical optics is the simplest model of light propagation. This model applies when an optical system's components are much larger than the wavelength of light.

A ray can be thought of as the path that light takes at the center of a slowly diverging electromagnetic beam of small lateral extent compared to the optical components in an optical system. A ray that travels in a slight inclination to the optical axis is called a paraxial ray.

Ray optics deals with the location and direction of light rays and the redirection by an optical component. It is well known that, in an optical system, paraxial ray propagation can be characterized by a 2 x 2 matrix called the ray-transfer matrix[9,10,11].

Matrix optics is a formal method of applying the ray-transfer matrix to characterize a paraxial optical system. Each component of this system has its own ray-transfer matrix. When tracing the ray through this system, the ray-transfer matrix of the system is the product of the individual component's matrix. The output ray's location and direction can be found with relative ease, even in complex optical systems, in the paraxial approximation.

Development of the Ray-Transfer Matrix, **T**.

We will develop ray-transfer matrices for only three special optical components: the Homogeneous Dielectric; the "Thin" Lens; and the Spherical Mirror. The first and last elements are specifically used in the simple Spherical Mirror Resonator.

We begin by considering only paraxial rays in an optical system where the slope of the ray r' equals the angle measured with respect to the optical axis.

Homogeneous Dielectric Ray Propagation. Figure 2 shows the initial position r_{in} and slope r'_{in} and the output position r_{out} and slope r'_{out} of a ray passing through a Homogeneous Dielectric.

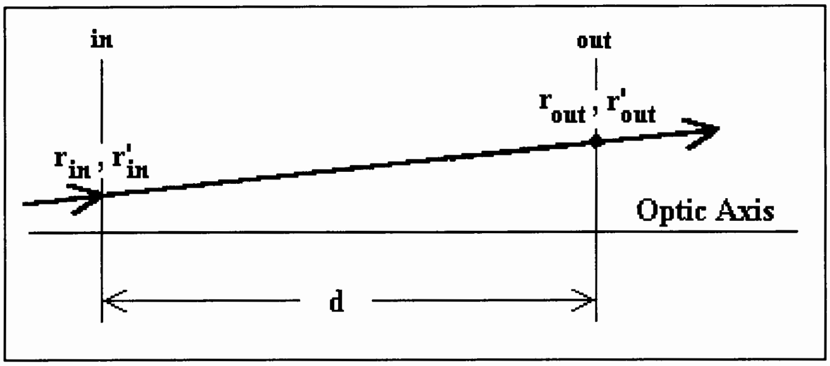


Figure 2. Ray propagation through a Homogeneous Dielectric of length d.

The output and input rays are related by:

$$r_{out} = 1 \cdot r_{in} + d \cdot r'_{in} = A \cdot r_{in} + B \cdot r'_{in} \tag{1}$$

$$r'_{out} = 0 \cdot r_{in} + 1 \cdot r'_{in} = C \cdot r_{in} + D \cdot r'_{in} \tag{2}$$

or

$$\begin{bmatrix} r_{out} \\ r'_{out} \end{bmatrix} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} r_{in} \\ r'_{in} \end{bmatrix} . \tag{3}$$

The homogeneous dielectric ray-transfer matrix $\mathbf{T_{HD}}$ is

$$\mathbf{T}_{HD} = \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} . \tag{4}$$

Note that $\mathbf{T}_{HD}$ is unimodular, i.e. AD - BC = 1.

"Thin" Lens Ray Propagation. Figure 3 shows two cardinal rays r_a and r_b passing through a "thin" Lens whose thickness is negligible.

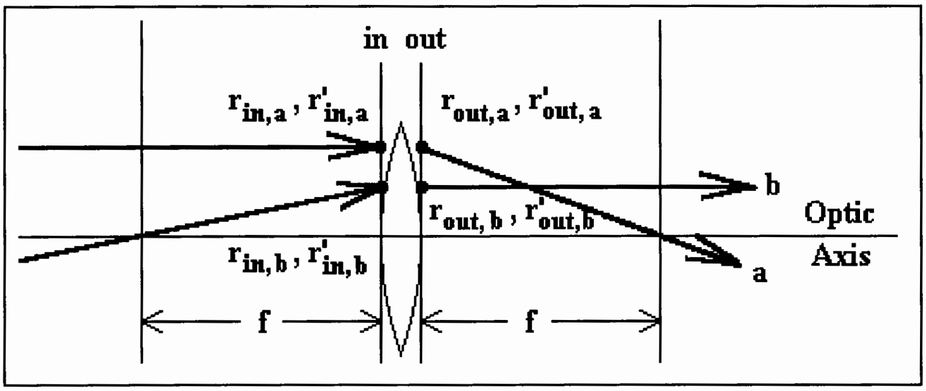


Figure 3. Ray propagation through a "Thin" Lens.

The output position equals the input position regardless of input direction, thus

$$r_{out} = r_{in} = A \cdot r_{in} + B \cdot r'_{in} \ . \tag{5}$$

Therefore A = 1 and B = 0.

According to geometrical optics a ray parallel to the optic axis will pass through the back focal point f as indicated by ray a in Figure 3. Here $r'_{in,a} = 0$ and $r'_{out,a} = -r_{in,a}/f$ so that

$$r'_{out,a} = -r_{in,a}/f = C \cdot r_{in,a} + D \cdot r'_{in,a} = C \cdot r_{in,a} + D \cdot 0 \ \text{ or } C = -1/f . \tag{6}$$

Ray b passes through the front focal point f and exits parallel to the optic axis. Here $r'_{out,b} = 0$ so that

$$r'_{out,b} = 0 = -1/f \cdot r_{in,b} + D \cdot r'_{in,b} \quad \text{and} \quad r'_{in,b} = r_{in,b}/f . \tag{7}$$

Therefore D = 1 and the ray-transfer matrix of a "Thin" Lens $\mathbf{T}_{TL}$ is

$$\mathbf{T}_{TL} = \begin{bmatrix} 1 & 0 \\ -1/f & 1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{8}$$

Note that $\mathbf{T}_{TL}$ is unimodular.

Spherical Mirror Ray Propagation. Figure 4 shows an incident and reflected ray upon a Spherical Mirror with radius of curvature R.

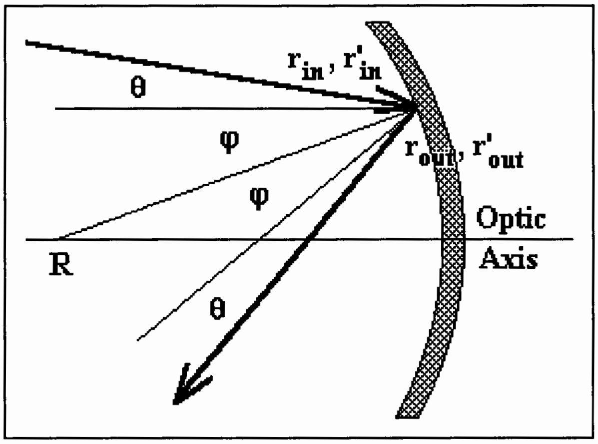


Figure 4. A Ray being reflected on a Spherical Mirror.

The input and output positions are the same so that $r_{out} = r_{in} = A \cdot r_{in} + B \cdot r'_{in}$. Therefore A = 1 and B = 0. For the input and output slopes we have

$$r'_{in} = -\tan(\theta) \approx \theta \quad \text{and} \quad \sin(\varphi) = r_{in}/R \approx \varphi \tag{9}$$

and

$$r'_{out} = -\tan(2\varphi + \theta) \approx -(2\varphi + \theta), \text{ or} \tag{10}$$

$$r'_{out} = C \cdot r_{in} + D \cdot r'_{in} = -2/R \cdot r_{in} + 1 \cdot r'_{in} . \tag{11}$$

Here C = -2/R and D = 1 so that the spherical mirror ray-transfer matrix $\mathbf{T}_{SM}$ becomes

$$\mathbf{T}_{SM} = \begin{bmatrix} 1 & 0 \\ -2/R & 1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{12}$$

Again note that $\mathbf{T}_{SM}$ is unimodular.

It is readily seen from the ray-transfer matrices $\mathbf{T}_{TL}$ and $\mathbf{T}_{SM}$ that reflection from a Spherical Mirror with radius of curvature R is equivalent, except the folding of the ray's path, to passage through a "Thin" Lens of focal length f = R/2.

Figure 5 shows an important use of the ray-transfer matrix: that of cascading optical elements together into a single optical component of ray-transfer matrix, $\mathbf{T}_{SYS}$.

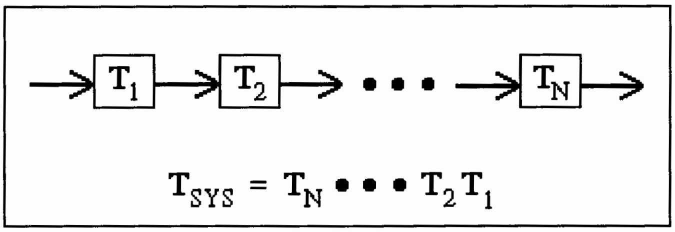


Figure 5. The cascading of Ray-Transfer Matrices.

The order of the above matrix multiplication is such that the incident ray's transfer matrix is placed to the right. This is analogous to the use of the "S" parameters in Microwave Circuit Theory.

THE STABLE SPHERICAL MIRROR RESONATOR

We will now develop the simple Spherical Mirror Resonator using the Homogeneous Dielectric and "Thin" Lens ray-transfer matrices; ultimately defining the Stability Diagram of Figure 7. The confinement condition of light rays within this resonator will be derived from two perspectives; the unbounded lens waveguide method and the self-consistent method.

The Unbounded Lens Waveguide Method

In this method we transform the spherical mirror system into a lens waveguide and analyze the light rays' paths as they traverse the periodic sequence. Figure 6 shows an empty laser cavity with its equivalent lens waveguide comprised of an unbounded

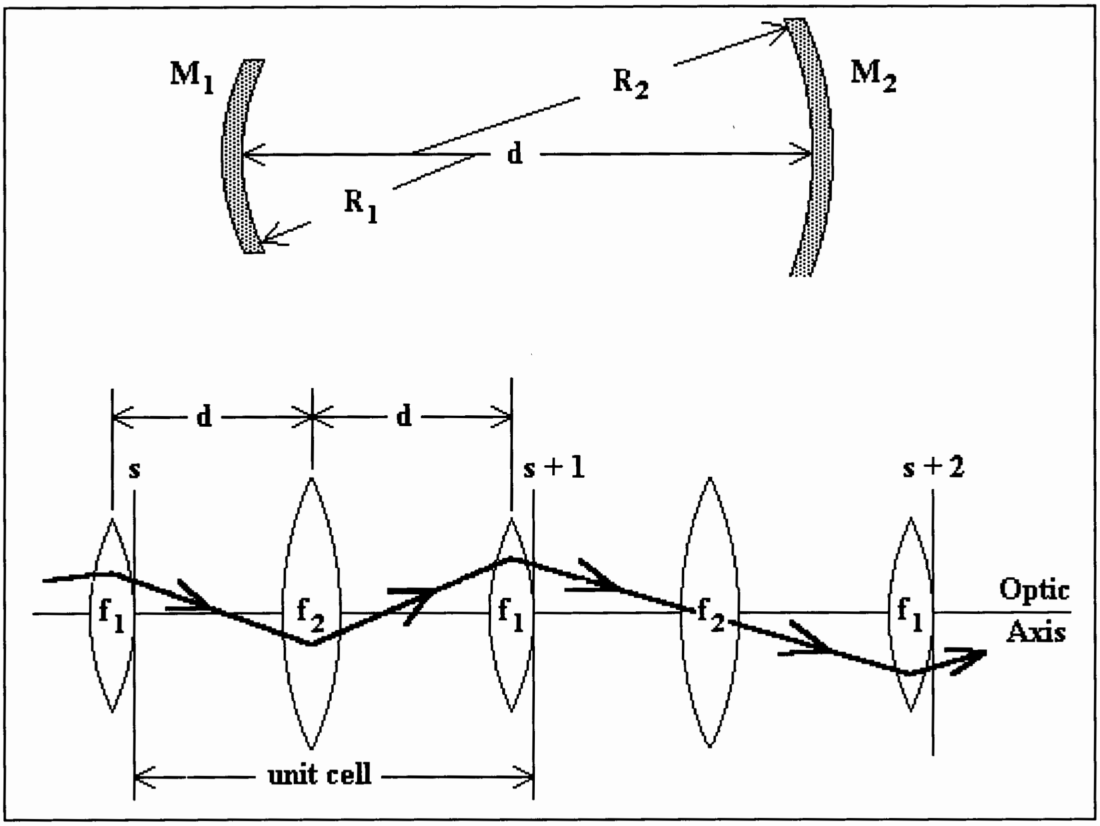


Figure 6. An empty laser cavity with its equivalent biperiodic lens sequence.

biperiodic lens sequence. We begin by making use of the ray-transfer matrices previously derived.

The ray-transfer of the unit cell is comprised of

$$\mathbf{T}_{\text{unit cell}} = \mathbf{T}_{\text{TL},1} \bullet \mathbf{T}_{\text{HD}} \bullet \mathbf{T}_{\text{TL},2} \bullet \mathbf{T}_{\text{HD}} \tag{13}$$

or

$$\mathbf{T}_{\text{unit cell}} = \begin{bmatrix} 1 & 0 \\ -1/f_1 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & 0 \\ -1/f_2 & 1 \end{bmatrix} \bullet \begin{bmatrix} 1 & d \\ 0 & 1 \end{bmatrix} \tag{14}$$

or finally

$$\mathbf{T}_{\text{unit cell}} = \begin{bmatrix} 1\text{-}d/f_2 & d(2 - d/f_2) \\ (d/f_1 - 1)/f_2 - 1/f_1 & (1\text{-}d/f_1)(1 - d/f_2) - d/f_1 \end{bmatrix}. \tag{15}$$

Consider the planes denoted by s, s + 1, s + 2, · · · in Figure 6. Ray propagation from one plane to the next can be written as

$$\begin{bmatrix} r_{s+1} \\ r'_{s+1} \end{bmatrix} = \mathbf{T}_{\text{unit cell}} \bullet \begin{bmatrix} r_s \\ r'_s \end{bmatrix} \tag{16}$$

or

$$r_{s+1} = A \cdot r_s + B \cdot r'_s \quad => \quad r'_s = (r_{s+1} - A \cdot r_s)/B \tag{17}$$

and

$$r'_{s+1} = (r_{s+2} - A \cdot r_{s+1})/B = C \cdot r_s + D \cdot r'_s \ . \tag{18}$$

Substituting r'_s we obtain

$$(r_{s+2} - A \cdot r_{s+1})/B = C \cdot r_s + (r_{s+1} - A \cdot r_s) \cdot D/B \ . \tag{19}$$

Combining terms and using AD - BC = 1 yields

$$r_{s+2} - (A + D) \cdot r_{s+1} + r_s = 0 \tag{20}$$

or

$$r_{s+2} - 2br_{s+1} + r_s = 0 \ \text{, where } b = (A + D)/2 = (1 - d/f_2 - d/f_1 + d^2/(2f_1f_2)) \ . \tag{21}$$

This last equation is in equivalent form to the differential equation r'' + kr = 0 which has solutions $r(z) = \rho \exp[\pm i(k)^{1/2}z]$. We are thus led to try a solution in the form of $r_s = \rho e^{is\theta}$ that when substituted into the last equation yields

$$e^{2i\theta} - 2be^{i\theta} + 1 = 0 \ . \tag{22}$$

Thus $e^{i\theta} = b \pm i(1 - b^2)^{1/2}$ so that $b^2 \leq 1$ and $\cos(\theta) = b$. The general solution is a linear combination of the form

$$r_s = \rho e^{is\theta} + \rho^* e^{-is\theta} \quad \text{or} \quad r_s = r_{max}\sin(s\theta + \alpha) \ . \tag{23}$$

The condition for ray confinement is such that θ be a real number so that the ray radius r_s oscillates as a function of the cell number s between r_{max} and $-r_{max}$. This means that $b^2 \leq 1$. A confined ray leads to a stable laser cavity.

There can also be the case when $b^2 > 1$. This has solutions in the form of $r_s = ce^{s\theta} + de^{-s\theta}$, where $e^{\pm\theta} = b \pm (b^2 - 1)^{1/2}$. Since the magnitude of either $e^{+\theta}$ or $e^{-\theta}$ exceeds unity, the ray radius will increase as a function of (distance) s. The ray is unconfined which leads to an unstable laser cavity.

In terms of system parameters, $b^2 \leq 1$ or $|b| \leq 1$ can be written as

$$-1 \leq (1 - d/f_2 - d/f_1 + d^2/\{2f_1f_2\}) \leq 1 \tag{24}$$

or

$$0 \leq (1 - d/(2f_1))\,(1 - d/(2f_2)) \leq 1\ . \tag{25}$$

When substituting $f_1 = R_1/2$ and $f_2 = R_2/2$ we obtain the confinement condition for simple spherical mirror resonators,

$$0 \leq (1 - d/R_1)\,(1 - d/R_2) \leq 1\ . \tag{26}$$

Figure 7 shows a graphic representation of the confinement condition given in the above equation [12]. The shaded areas represent high diffraction loss where $b^2 > 1$. Here the beam is not well confined and spills over the mirror's edge. Whereas in the confined areas the beam satisfies $b^2 \leq 1$ and a low loss condition occurs for optical resonance.

Note that when the mirror curvatures are equal ($R_1 = R_2$) the system is symmetrical and lies along the diagonal line depicted by the progression of 1 to 6 in Figure 7. These various configurations are shown in Figure 8.

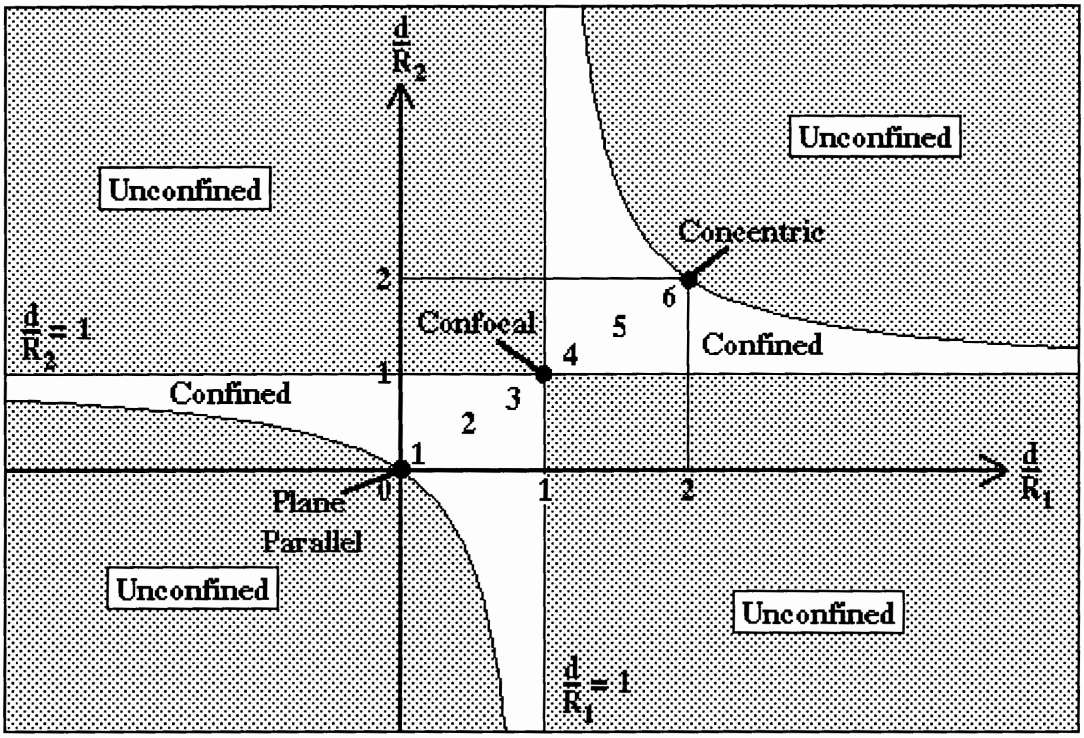


Figure 7. The Beam Confinement (or Stability) Diagram for optical resonators.

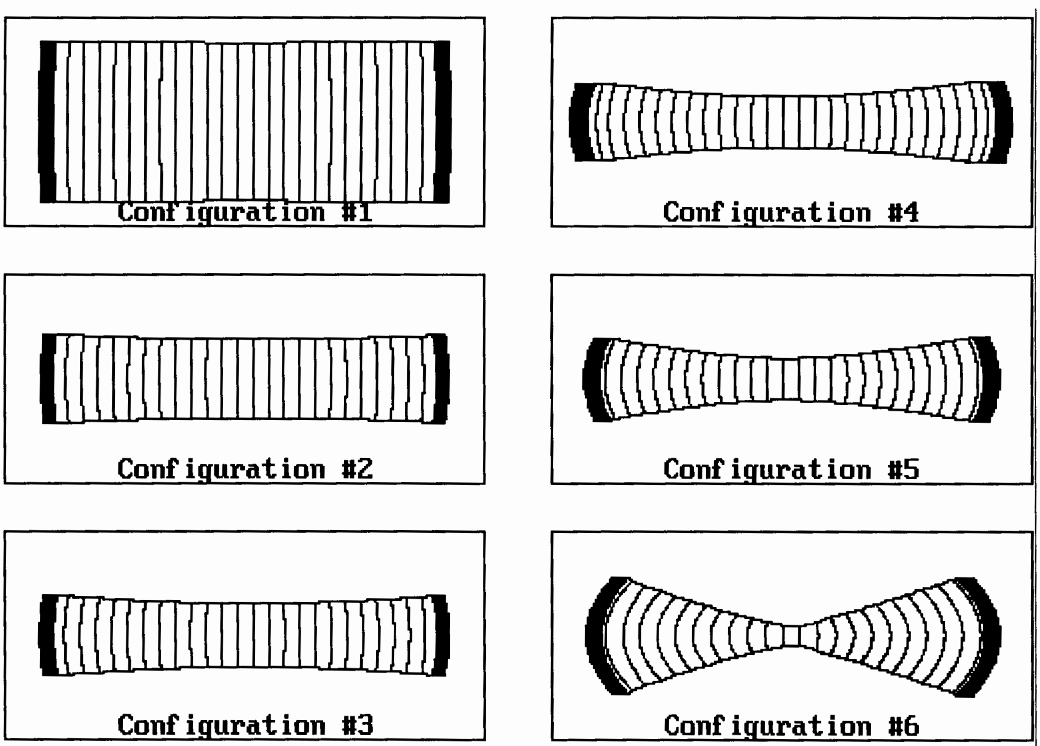


Figure 8. A progressive selection of Symmetric Mirror curvatures taken from Figure 7.

The Gaussian Beam. The Gaussian beam wavefronts, shown in each configuration of Figure 7, all have the same curvature as that of the radius of curvature of their respective mirrors. The beam is reflected back on itself and will retrace its path back and forth within the resonator. The beam then can exist self-consistently within the cavity satisfying the Helmholtz equation ($\nabla^2\mathbf{E} + k^2(\mathbf{r})\mathbf{E} = 0$) as well as the boundary conditions imposed by the mirrors. The Gaussian beam is a mode of the spherical-mirror resonator provided that its phase also retraces itself.

The Fundamental Gaussian beam is given by

$$E(x,y,z) = E_o \left[\frac{w_o}{w(z)}\right] \exp\left[-\frac{r^2}{w^2(z)} - i\left\{kz + \frac{kr^2}{2R(z)} - \eta(z)\right\}\right], \tag{27}$$

where $E(x,y,z) =$ the Electric Field,

E_o = the initial amplitude,

w_o = the waist radius. The waist radius w_o is called the spot size,

$w(z)$ = the radial distance r at which the field amplitude is down by a factor of $1/e$ compared to its value on the z axis,

r = $(x^2 + y^2)^{1/2}$; the radial distance,

i = $(-1)^{1/2}$; the imaginary number,

$\eta(z)$ = the Guoy phase shift = the phase retardation relative to a plane wave,

k = $2\pi/(n\lambda)$; the wavenumber and n = the index of refraction, and

$R(z)$ = the radius of curvature of the wavefronts.

Some of these parameters are defined as

$$w^2(z) = w_o^2[1 + (\lambda z/(\pi w_o^2 n))^2] = w_o^2[1 + (z/z_o)^2] \tag{28}$$

$$R(z) = z[1+ (\pi w_o^2 n/(\lambda z))^2] = z[1 + (z_o/z)^2] \tag{29}$$

$$\eta(z) = \tan^{-1}[\lambda z/(\pi w_o^2 n)] = \tan^{-1}(z/z_o) \tag{30}$$

$$z_o \equiv \pi w_o^2 n/\lambda \tag{31}$$

where z_o = the confocal parameter having the following properties at the $z = z_o$ plane:

a) the intensity on the beam axis is ½ the $z = 0$ peak value;

b) the beam radius is $(2)^{1/2}$ larger than w_o. (the beam area is doubled vs. at $z=0$);

c) the phase on the beam axis is retarded by 90 degrees to that of a plane wave; and

d) the radius of curvature is at its smallest value, $R_{min} = 2z_o$.

The confocal parameter, sometimes known as the depth of focus, is a convenient measure of the divergence of an output beam. It is also an estimate of where Fresnel diffraction ends ($z < z_o$) and where Fraunhofer diffraction begins ($z > z_o$).

<u>The Self Consistent Method</u>

In this method of determining the confinement condition we will make use of the complex beam radius q(z) which enables one to determine the beam radius w(z) and its radius of curvature R at any z plane. It is defined as

$$\frac{1}{q(z)} = \frac{1}{R(z)} - \frac{i\lambda}{n\pi w^2(z)} . \tag{32}$$

<u>The ABCD Law</u>. The usefulness of the q parameter is found when applying the ABCD law where

$$q_{out} = \frac{Aq_{in} + B}{Cq_{in} + D} = \text{The ABCD Law.} \tag{33}$$

A, B, C and D are the elements of the transfer matrix **T** and the output and input Gaussian beams are characterized by q_{out} and q_{in} respectively. Full characterization requires additional knowledge of the beam axis and intensity.

Gaussian beam propagation through a complex arbitrary paraxial optical system can be determined if one knows either q_{in} or q_{out} and the system's transfer matrix $\mathbf{T}_{SYS}$. The beam radius of curvature R(z) and waist w(z) at any z plane can then be recovered according to the above two equations.

We will now apply the ABCD law to a generalized resonator by what is called the self consistent method. A stable resonant eigenmode is one which reproduces itself after one round trip. An arbitrary reference plane is selected and the ABCD elements for one *complete* round trip are then used in the ABCD law. At the reference plane the complex beam parameter $q = q_{in} = q_{out}$ if the beam is to reproduce itself. We require that $q = (Aq + B)/(Cq + D)$. Solving for $1/q$ using $AD - BC = 1$ yields

$$1/q = [(D - A) \pm i(4 - (D + A)^2)^{1/2}]/2B. \tag{34}$$

Since $1/q$ must be complex, due to the waist being finite size, we have $4 - (D + A)^2 > 0$ or $|(A + D)/2| \leq 1$. This is the confinement condition earlier denoted as $|b| \leq 1$. The radius of curvature R and the waist w at the reference plane are

$$R = 2B/(D - A) \text{ and} \tag{35}$$

$$w = (\lambda/\pi n)^{1/2} |B|^{1/2}/[1 - ((D + A)/2)^2]^{1/4} . \tag{36}$$

The Paraxial Wave Equation

The paraxial wave equation is an approximation to the scalar wave equation which is derived from Maxwell's equations in free space. We begin with the scalar wave equation in the form

$$[\nabla^2 + k^2]E(x,y,z) = 0 , \tag{37}$$

where $E(x,y,z)$ is the phasor amplitude of a field distribution that is sinusoidal in time. The flow of energy is predominantly along a single direction, the z axis. The primary spatial dependence of $E(x,y,z)$ will be an $\exp(-ikz)$ variation which has a spatial period of one wavelength λ in the z direction. The transverse variations due to diffraction and propagation are usually slow compared one optical cycle as in the plane-wave $\exp(-ikz)$ variation. To get better resolution of the transverse dependence we write $E(x,y,z)$ in the form

$$E(x,y,z) = \psi(x,y,z)e^{-ikz} , \tag{38}$$

where $\psi(x,y,z)$ = a complex scalar wave amplitude which describes the transverse profile of the beam. Substituting this into the scalar wave equation yields, in Cartesian coordinates, the reduced equation

$$\frac{\partial^2\psi}{\partial^2 x} + \frac{\partial^2\psi}{\partial^2 y} + \frac{\partial^2\psi}{\partial^2 z} - 2ik\frac{\partial\psi}{\partial z} = 0. \tag{39}$$

The z dependence in the transverse direction is assumed slow enough that

$$\left|\frac{\partial^2\psi}{\partial^2 z}\right| \ll \left|2k\frac{\partial\psi}{\partial z}\right|. \tag{40}$$

This is the *slowly varying envelop approximation* or *paraxial approximation.* By dropping the second partial derivative in z, the exact scalar wave equation becomes the paraxial wave equation

$$\frac{\partial^2\psi}{\partial^2 x} + \frac{\partial^2\psi}{\partial^2 y} - 2ik\frac{\partial\psi}{\partial z} = 0\,. \tag{41}$$

More generally this equation becomes

$$\nabla^2{}_t\psi(s,z) - 2ik\frac{\partial\psi(s,z)}{\partial z} = 0, \tag{42}$$

where s denotes either the x,y or r,θ coordinates in rectangular or cylindrical coordinates respectively and $\nabla^2{}_t$ is the laplacian operator operating on these coordinates in the transverse plane.

OPTICAL RESONATOR ALGEBRA

We now will determine the Gaussian beam whose curvature matches the mirror curvatures R_1 and R_2 at the location of the mirrors M_1 and M_2 respectively. We will find the confocal parameter z_0 and the waist w_0 from the system's parameters (d, R_1 and R_2). Once z_0 and w_0 are known the Gaussian beam is thus defined except for the initial amplitude E_0. The beam direction is taken along the z axis. The location of w_o is where z=0. The locations of M_1 and M_2 are where $z=z_1$ and $z=z_2$ respectively.

To determine the waist radii at the mirrors of given R_1 and R_2 we first find z_0. Then the waist w_0 is found. Finally we calculate the waist radii w_1 and w_2. We begin with the equations

$$R_i = z_i[1+(z_0/z_i)^2] \quad => \quad z_i = R_i/2 \pm (R_i^2 - 4z_0^2)/2; \; i=1,2 \tag{43}$$

and

$$d=z_2 - z_1 \, . \tag{44}$$

Through-out this section on resonator algebra, the mirror curvature R_1 or R_2 is positive if the center of curvature is to the left of the mirror and negative otherwise. Solving for z_0 we have

$$z_0 = [-d(R_1 + d)(R_2 - d)(R_2 - R_1 - d)/(R_2 - R_1 - 2d)^2]^{1/2} \, . \tag{45}$$

The waist is given by

$$w_0 = (\lambda z_0/\pi n)^{1/2} \, . \tag{46}$$

The waist radii at the mirrors M_1 and M_2 are given by

$$w_i = w_0 \, [1 + (z_i/z_0)^2]^{1/2}.; \; i=1,2 \, . \tag{47}$$

The following discussion will involve the Symmetrical Mirror Resonator (R_1=R_2). Three special cases will be investigated: the Confocal Resonator; the Concentric Resonator; and the Plane-Parallel Resonator.

The Symmetric Mirror Resonator

This family of resonators lie along the diagonal line depicted in the Stability Diagram of Figure 7 by the linear progression from 1 to 6. This is where both mirrors are identical both being concave ($R=R_1=R_2 >0$).

We must remember to redefine the radius of mirror curvature R_1 or R_2 as positive if the center of curvature is to the left of the mirror and negative otherwise. Therefore we put $R=-R_1=R_2$ in the above equation for z_0 to yield

$$z_0 = [d(2R - d)]^{1/2}/2 \, . \tag{48}$$

The waist is given by

$$w_0 = (\lambda z_0/\pi n)^{1/2} = (\lambda/\pi n)^{1/2}[(Rd - d^2/2)/2]^{1/4} . \tag{49}$$

The waist radii at the mirrors are given by

$$w_i = w_0 [1 + (z_i/z_0)^2]^{1/2}. = (\lambda d/2\pi n)^{1/2}[2R^2(d(R - d/2))]^{1/4} ; i=1,2 \tag{50}$$

where $z_1 = -d/2$ and $z_2 = d/2$.

The Confocal Resonator. This is a special symmetrical resonator where the radii of curvature of both concave mirrors equals the cavity length d. Figure 9 depicts this type of resonator.

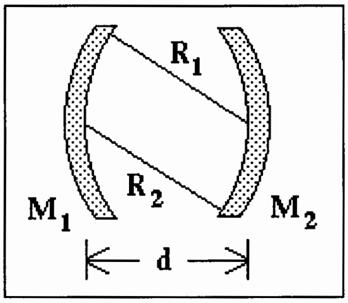


Figure 9. The Confocal Resonator where R_1=R_2=d.

Here we put R = d into the above equation for z_0 and obtain

$$z_0 = [d(2R - d)]^{1/2}/2 \Rightarrow (z_0)_{conf} = d/2 . \tag{51}$$

The waist at z=0 becomes

$$(w_0)_{conf} = (\lambda(z_0)_{conf}/\pi n)^{1/2} = (\lambda d/2\pi n)^{1/2} . \tag{52}$$

The waist at the mirrors becomes

$$(w_i)_{conf} = (w_0)_{conf} [1 + (z_i/(z_0)_{conf})^2]^{1/2} = (2)^{1/2}(w_0)_{conf} = (\lambda d/\pi n)^{1/2} ; i=1,2 \tag{53}$$

where $z_1 = -z_0$ and $z_2 = z_0$.

In the Confocal Resonator the waist radii $(w_{1,2})_{conf}$ is at the minimum value.

The Concentric Resonator. This is another special symmetrical resonator where the center of curvature of each concave mirror coincides. Figure 10 illustrates this resonator.

We put R=d/2 into the equation for z_0 and obtain

$$z_0 = [d(2R - d)]^{1/2}/2 \;\Rightarrow\; (z_0)_{conc} = 0 \,. \tag{54}$$

The waist at z=0 becomes

$$(w_0)_{conc} = (\lambda(z_0)_{conc}/\pi n)^{1/2} = 0 \,. \tag{55}$$

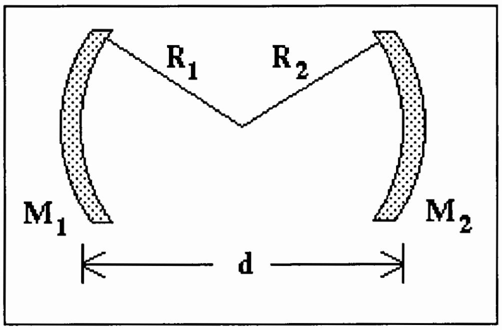


Figure 10. The Concentric Resonator where R_1=R_2=d/2.

The waist at the mirrors becomes

$$(w_i)_{conc} = (w_0)_{conc}\,[1 + (z_i/(z_0)_{conc})^2]^{1/2} = \infty \;;\; i=1,2 \tag{56}$$

where z_1 = -d/2 and z_2 = d/2.

In the Concentric Resonator the waist radii $(w_{1,2})_{conc}$ is at the maximum value and with $(w_0)_{conc}$ = 0 implying a maximum beam divergence. This is analogous to a spherical wave. The Concentric Resonator is on the border line of the confined and unconfined regions of the Confinement Diagram.

The Plane-Parallel Resonator. This is another special symmetrical resonator where the radii of curvature of both plane mirrors equal infinity. Figure 11 illustrates this type of resonator.

We put R=∞ into the equation for z_0 and obtain

$$z_0 = [d(2R - d)]^{1/2}/2 \;\Rightarrow\; (z_0)_{plane} = \infty \,. \tag{57}$$

The waist at z=0 becomes

$$(w_0)_{plane} = (\lambda(z_0)_{plane}/\pi n)^{1/2} = \infty \,. \tag{58}$$

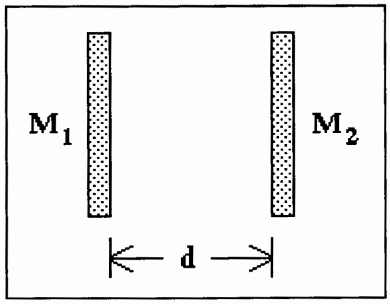


Figure 11. The Plane-Parallel Resonator where $R_1=R_2=\infty$.

The waist at the mirrors becomes

$$(w_i)_{plane} = (w_0)_{plane} [1 + (z_i/(z_0)_{plane})^2]^{1/2} = \infty \ ; i=1,2 \tag{59}$$

where $z_1 < 0$ and $z_2 > 0$

In the Plane-Parallel Resonator the waist radii $(w_{1,2})_{plane}$ is also at the maximum value and with the waist $(w_0)_{plane} = \infty$ implying a minimum beam divergence. This is analogous to a plane wave. The Plane-Parallel Resonator is on the border of the confined and unconfined regions of the Confinement Diagram.

It is helpful to plot $(\pi w^2/\lambda d)^{1/2}$ vs. d/R to get a feel for the way the waist at the mirrors vary with d/R by keeping the mirror curvatures constant while changing the mirror spacing. A plot of this function is shown in Figure 12.

The Half-Symmetric Resonator. In this resonator one of the mirrors is plane ($R = \infty$) and the other is concave ($R = R_2$). The allowed value for d/R_2 must be between 0 and 1 as can be seen in the Confinement Diagram of Figure 6. This resonator can be transformed into a symmetric resonator by substituting a mirror identical to the concave one for the plane mirror. The cavity length must then be doubled, i.e. d => 2d.

The beam waist is located on the plane mirror, say mirror M_1. In this case the expression for the waist radius at the plane and concave mirrors is given by

$$w_0 = w_1 = (\lambda d/\pi n)^{1/2}[(1-d/R_2)/(d/R_2)]^{1/4} \tag{60}$$

and

$$w_2 = (\lambda d/\pi n)^{1/2}[1/((1-d/R_2)(d/R_2))]^{1/4} . \tag{61}$$

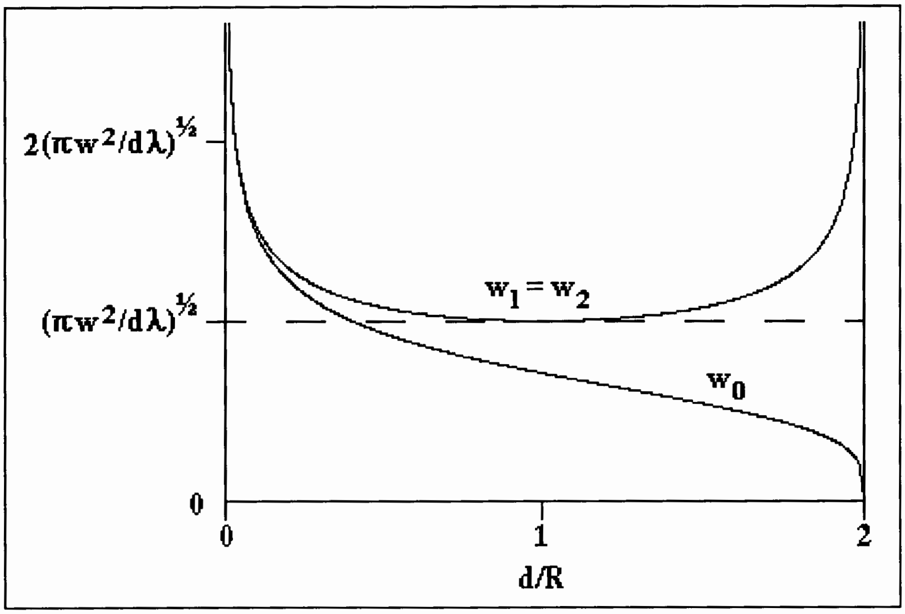


Figure 12. The $(\pi/\lambda d)^{1/2}$ scaled beam radius at the waist, w_0 and at the mirrors, w_1=w_2, for a stable Symmetric Resonator as a function of d/R.

Again we plot $(\pi w^2/\lambda d)^{1/2}$ vs. d/R_2 and see how the beam waist varies as one keeps the mirror curvature constant while changing the mirror distance. This is shown in Figure 13.

HIGHER-ORDER TRANSVERSE MODES

Modes of a resonator are different intensity distributions that retrace themselves when reflected between the resonator mirrors. Each mode's wave front matches the curvature of the mirrors at the mirrors.

We will discuss the Hermite-Gaussian beam which is used in rectangular geometry and the Laguerre-Gaussian beam which is used in cylindrical geometry.

Hermite-Gaussian Modes

These modes are the most widely used for the complete solution set to the paraxial wave equation since most lasers exhibit x,y astigmatism. Siegman shows that by solving

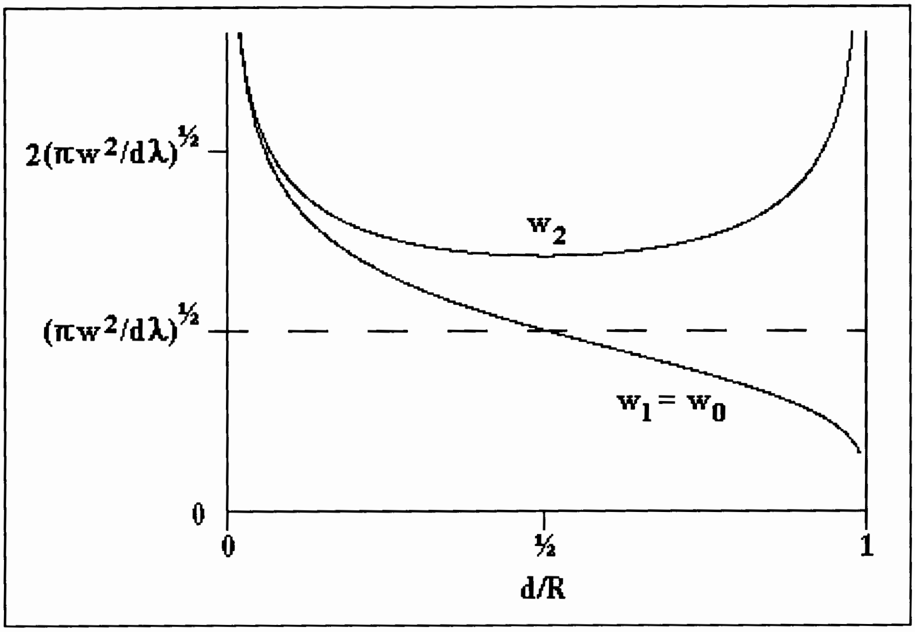


Figure 13. The $(\pi/\lambda d)^{1/2}$ scaled beam radius at the waist, $w_0 = w_1$ at the Plane Mirror and at the Concave Mirror, w_2, for a stable Half Symmetric Resonator as a function of d/R.

the paraxial wave equation in Cartesian coordinates one obtains the Hermit-Gaussian modes [13]. The result is

$$E(x,y,z)_{n,m} = (E_o)_{n,m} \left[\frac{w_o}{w(z)} \right] H_n \left[\frac{(2)^{1/2}x}{w(z)} \right] H_m \left[\frac{(2)^{1/2}y}{w(z)} \right]$$

$$\times \exp\left[- \frac{x^2+y^2}{w^2(z)} - i\{kz + k\left[\frac{x^2+y^2}{2R(z)} \right] - (n+m+1)\eta(z)\right] , \qquad (62)$$

where the integers $n,m \geq 0$ are the x,y modal indices respectively and the functions H_n and H_m are the Hermite polynomials of order n,m respectively. All other variables are as previously defined. These modes exhibit x,y symmetry about the x,y modal axes respectively.

Resonance Frequencies of the Hermite-Gaussian Modes. The phase of the (n,m) mode on the beam axis is the imaginary component of the E-Field. From the above equation we have

$$\varphi(x=0,y=0,z) = kz - (n+m+1)\eta(z) . \qquad (63)$$

The phase delay of a beam in a *complete* round trip in a resonator of length d must be set to a multiple of 2π in order that the beam retrace itself [14]. Thus

$$2kd - 2(n + m + 1)\Delta\eta = 2\pi q \, , \tag{64}$$

where k = the wave number,

d = the cavity length,

n,m = the x,y modal indices,

$\Delta\eta = \eta(z_2) - \eta(z_1)$,

z_2, z_1 = M_2, M_1 mirror positions,

$\eta(z) = \tan^{-1}(z/z_0)$,

z_0 = the confocal parameter and,

q = the axial mode index = $0, \pm1, \pm2, \cdots$.

This leads to resonance frequencies of the Hermite-Gaussian modes defined by

$$\nu_{n,m,q} = c/(2d)[q + (n + m + 1)\Delta\eta/\pi] \, . \tag{65}$$

Modes of different q, but the same (n,m), have identical intensity distributions. They are called *longitudinal* or *axial* modes. The (n,m) modes refer to the transverse x,y dimensions and are called *transverse* modes.

These resonance frequencies satisfy the following properties:

a) Longitudinal modes corresponding to the transverse mode (n,m) have a resonance frequency spacing of $c/(2d) = \nu_{n,m,q+1} - \nu_{n,m,q}$;

b) All transverse modes, for which the sum of the indices n+m is the same, have the same resonance frequencies; and

c) Two transverse modes (n,m) and (n',m') corresponding to the same longitudinal mode q have resonance frequencies spaced by

$$\nu_{n,m,q} - \nu_{n',m',q} = c/(2d)[(n + m) - (n' + m')]\Delta\eta/\pi.$$

This expression determines the frequency shift between the sets of longitudinal modes of indices (n,m) and (n',m').

Laguerre-Gaussian Modes

Siegman also gives an equally valid set of complete solutions to the paraxial wave equation in cylindrical coordinates [15]. These Laguerre-Gaussian solutions have the form

$$E(r,\theta,z)_{p,m} = (E_o)_{p,m} \left[\frac{w_o}{w(z)} \right] \left[\frac{(2)^{1/2} r}{w(z)} \right]^m L_p^m \left[\frac{2r^2}{w^2(z)} \right]$$

$$\times \exp\left[- \left[\frac{r^2}{w^2(z)} \right] - i\left\{kz + k\left[\frac{r^2}{2R(z)} \right] - (2p + m + 1)\eta(z) + m\theta\right\}\right] , \tag{66}$$

where the integer $p \geq 0$ is the radial index and the integer $m = 0, \pm 1, \pm 2, \cdots$ is the azimuthal mode index; the L_p^m functions are the Laguerre polynomials; and all other quantities are as previously defined.

These modes exhibit cylindrical symmetry, with modes having circles of constant intensity in the radial direction and an $e^{im\theta}$ variation in the azimuthal direction. Alternately, linear combinations of the $\pm m$ terms can be formed to give $\cos(m\theta)$ and/or $\sin(m\theta)$ variations, leading to 2m nodal lines running radial outward from the mode axis.

Resonance Frequencies of the Laguerre-Gaussian Modes. Again, we proceed as in the Hermite-Gaussian case. The phase of the (p,m) mode on the beam axis is the imaginary component of the E-Field. Thus

$$\varphi(r{=}0,\theta,z) = kz - (2p + m + 1)\eta(z) - m\theta . \tag{67}$$

The phase delay of a beam in a *complete* round trip in a resonator of length d must be set to a multiple of 2π in order that the beam retrace itself. Thus

$$2kd - 2(2p + m + 1)\Delta\eta - 2m\theta = 2\pi q , \tag{68}$$

This leads to resonance frequencies of the Laguerre-Gaussian modes defined by

$$\nu_{p,m,q} = c/(2d)[q + (2p + m + 1)\Delta\eta/\pi + m\theta/\pi] . \tag{69}$$

Modes of different q, but the same (p,m), have identical intensity distributions. They are called *longitudinal* or *axial* modes. The (p,m) modes refer to the transverse r,θ dimensions and are called *transverse* modes.

These resonance frequencies satisfy the following properties:

a) Longitudinal modes corresponding to the transverse mode (p,m) have a resonance frequency spacing of $c/(2d) = \nu_{p,m,q+1} - \nu_{p,m,q}$;

b) All transverse modes, for which the sum of the indices p+m is the same, have the same resonance frequencies; and

c) Two transverse modes (p,m) and (p',m') corresponding to the same longitudinal mode q have resonance frequencies spaced by

$$\nu_{p,m,q} - \nu_{p',m',q} = c/(2d)[[(2p + m) - (2p' + m')]\Delta\eta/\pi + (m-m')\theta/\pi].$$

This expression determines the frequency shift between the sets of longitudinal modes of indices (p,m) and (p',m').

Since either solution set (Hermite-Gaussian or Laguerre-Gaussian) can be used, we must be able to expand the Hermite solutions in terms of Laguerre functions and vice versa.

Most laser systems incorporate rectangular geometry, such as Brewster's mirrors or tilted components, such that the beam elects to oscillate in near-Hermite-Gaussian modes. The work in this thesis solely has cylindrical symmetry with the azimuthal index $m = 0$. Therefore the beam elects to oscillate in Laguerre-Gaussian L_p modes only.

CHAPTER III

RESONATOR MIRRORS

In this chapter we will discuss the design of the following three types of resonator mirrors: 1) the Spherical Mirror, a well known type used in most laser systems; 2) the Fresnel Mirror, not used in laser systems probably due to it's manufacturability; and 3) the Tiered Fresnel Mirror, a novel type that can be efficiently manufactured using segments of the Integrated Circuits Process. These mirrors are shown in Figure 14.

Also discussed is diffraction losses in an optical resonator system. This is important because it is a figure of merit to compare the novel Tiered Fresnel Mirror to the Spherical Mirror.

THE SPHERICAL MIRROR

The design of the Spherical Mirror is simple. It has a constant radius of curvature R, where R equals the distance that is normal to the mirror's surface at the center of the mirror to it's center of curvature. This is shown in Figure 15. Also shown is the mirror's radial dimension r, where r equals the distance that is tangent to the mirror's surface at it's center to the edge.

Different mirrors, r and/or R being different, will naturally yield a different performance in a laser system. The mirror performance or diffraction loss associated with a resonator mirror is a function proportional to the resonator Fresnel number N, where

$$N \equiv r^2/(\lambda d) \ . \tag{70}$$

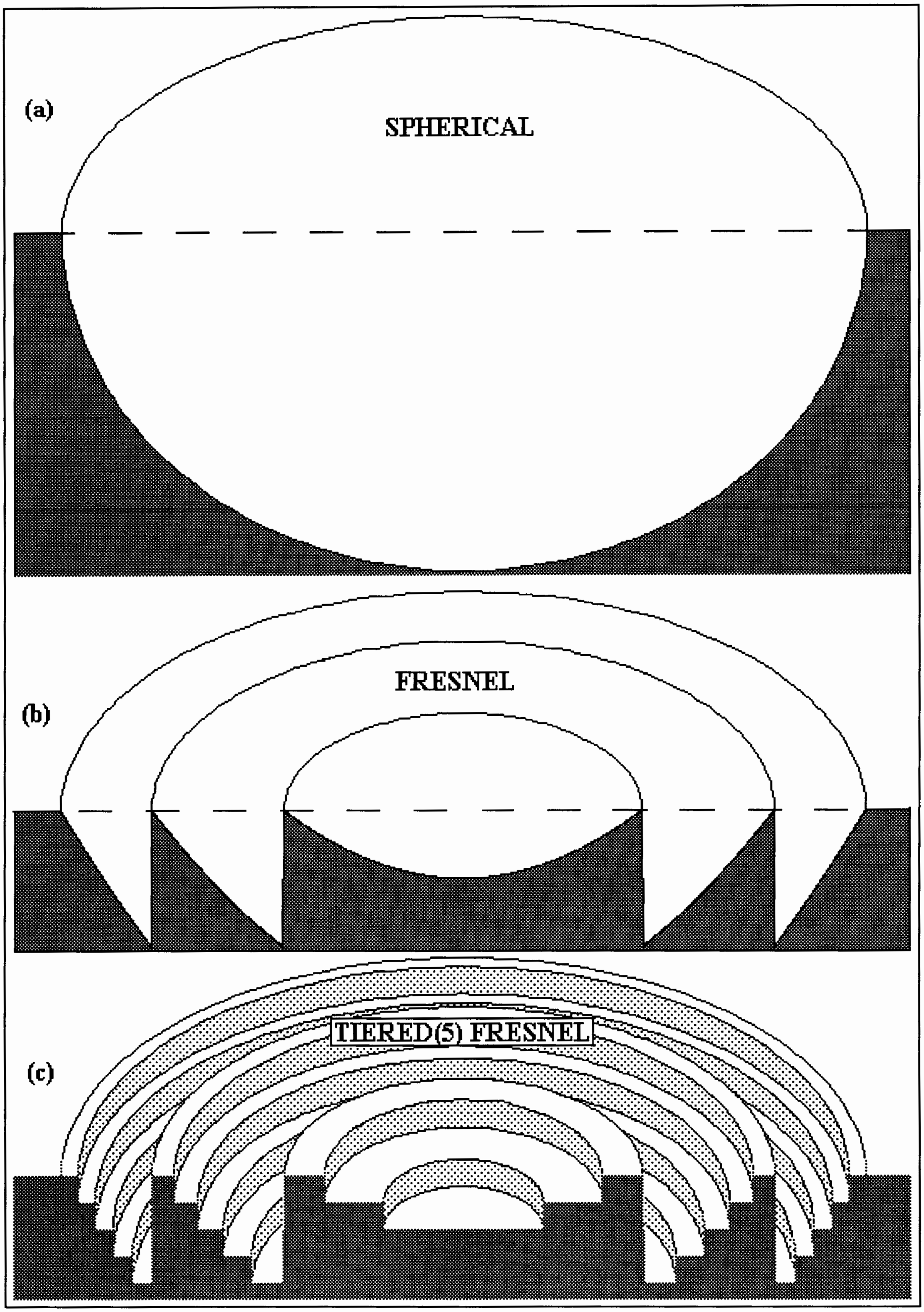


Figure 14. A 3d cross-sectional view of: a) the Spherical Mirror, b) the Fresnel Mirror, and c) the Tiered(5) Fresnel Mirror having 5 tiers per zone.

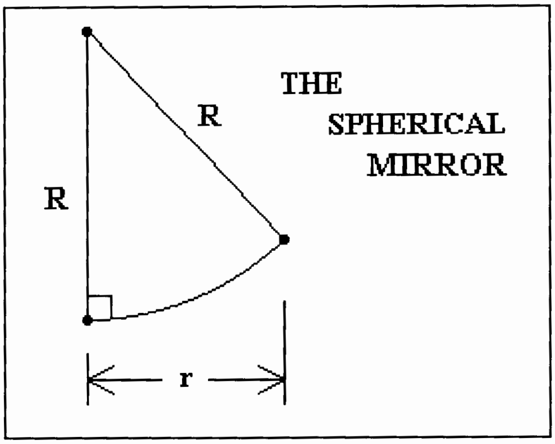


Figure 15. Critical points of the Spherical Mirror.

Here r is the mirror radius, d is the mirror separation or cavity length and λ is the wavelength of the laser light.

THE FRESNEL MIRROR

The Fresnel Mirror is essentially sections of a Spherical Mirror that are set side by side in a semi-planar manner as shown in Figure 16. Imagine drawing some concentric circles spaced $\lambda/2$ apart. Then draw a line tangent to one of the circles. The intersection of the line with the circles defines the planar direction of the mirror. Next we draw two parallel dashed lines a quarter of a wavelength above and below the solid line. Then the mirror is defined by tracing each circle between the dashed lines as shown in Figure 16 (The scale of Figures 13-15 is greatly skewed. In practice, the mirror curvature R is on the order of one meter whereas the mirror radius r is on the order of ten millimeters).

In actuality, the transition between circles is almost a vertical transition. Note that the transition distance is purposely selected to be a half a wavelength. Two reflected waves, one on each side of the transition, will have a phase difference equal to 2π, i.e. a wavelength. This is because one of the waves travels a half wavelength farther *before* being reflected as well as a half wavelength farther *after* the reflection. Thus this wave

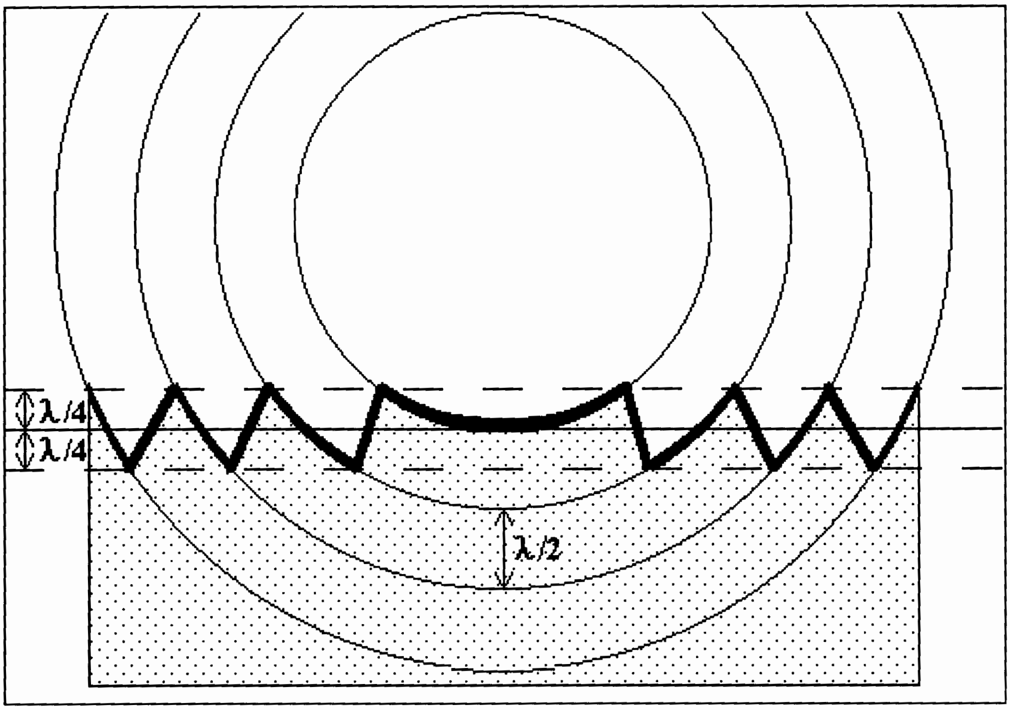


Figure 16. A cross-section of a Fresnel Mirror.

will lag the other in phase by 2π. Figure 17 shows the critical points in the design of the Fresnel Mirror.

The Fresnel Number N

The Fresnel number is an important number in the discussion of resonator mirrors with circular symmetry. It accounts for the mirror size which affects the diffraction loss or beam spill over at the edge and also the number of TEM modes allowed to oscillate without being quenched by diffraction losses. It is derived as follows [16].

Assume a plane wave is originated from a circular aperture as shown in Figure 18. The wave front is divided into a number of annular regions called Fresnel zones such that the boundaries are increments of $\lambda/2$. The Fresnel zone radii are defined from the figure as

$$r_N^2 = (d + N\lambda/2)^2 - d^2 \quad \text{or} \quad r_N^2 = Nd\lambda + (N\lambda/2)^2 . \tag{71}$$

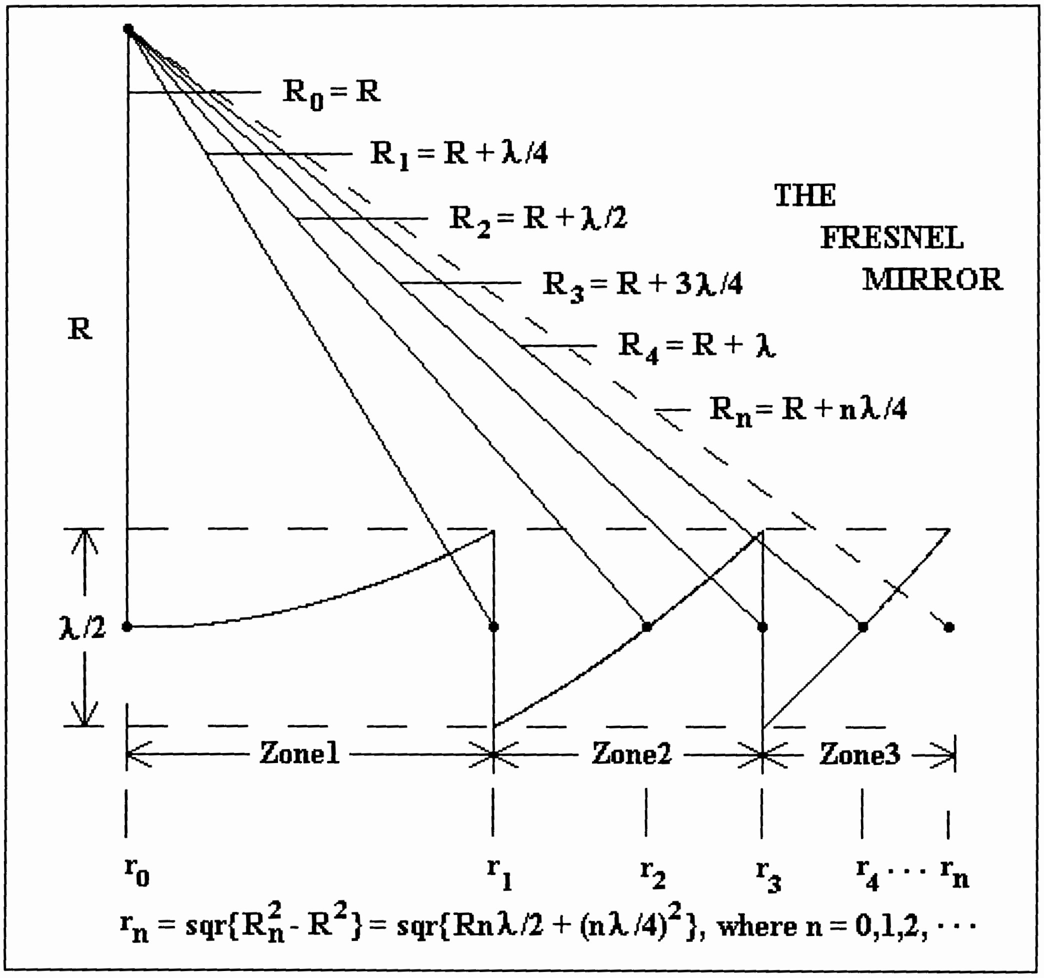


Figure 17. Critical points of the Fresnel Mirror. Shown is a radial plot of a Fresnel Mirror whose size spans the first three Fresnel zones.

With d>>λ and N not being extremely large, we can neglect the second term. Thus

$$r_N^2 = N\lambda d \quad \Rightarrow \quad N = r_N^2/\lambda d \,. \tag{72}$$

Here N is an integer yet in general N is a positive number.

We now ask the question, how many Fresnel zones are encompassed by the circular aperture? From the right triangle indicated by AOP in Figure 18, we have

$$N = r_N^2/\lambda d \,, \tag{73}$$

where N = the number of Fresnel zones within the aperture as seen from point P.

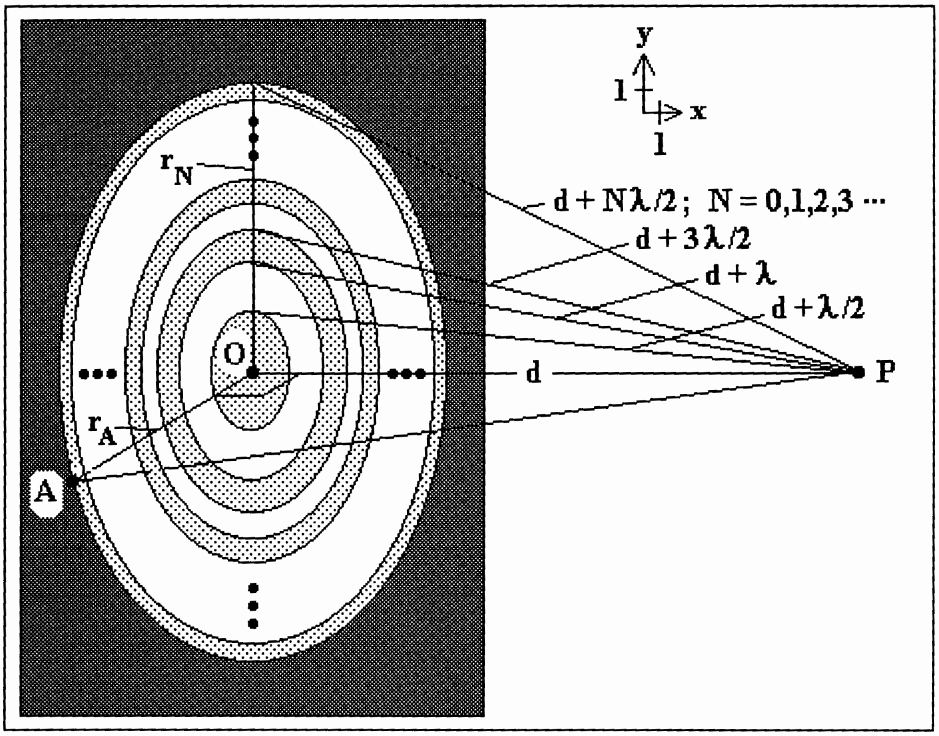


Figure 18. A circular aperture with a transmitted plane wave divided into annular rings. Each ring is a Fresnel zone with N equalling the Fresnel number.

The intensity at point P will rise periodically from zero to a maximum and back again to a minimum as the number of Fresnel zones within the aperture is increased from zero. This is because successive Fresnel zones tend to cancel each other. The resultant phase angle when radially traversing any Fresnel zone is equal to π radians.

For each additional Fresnel zone the vibration curve rotates one-half turn and a phase angle of π as it spirals inward. Thus when traversing any two adjacent zones, the resultant phase angle equals zero radians yet the resultant amplitude does not quite equal zero due to the obliquity factor.

If we replace the circular aperture with a reflective mirror, one can change the phase difference between odd and even zones so that both sets are in phase with each other as seen at the point P. This is accomplished by having a step height difference between the odd and even zones equal to $\lambda/4$. Note that the phase is not constant throughout any zone due to the planar structure. Ideally the phase should not change as in the case of the Spherical Mirror[17].

THE TIERED FRESNEL MIRROR

The Tiered Fresnel Mirror is an approximated version of the Fresnel Mirror. The degree of approximation is directly proportional to the number of tiers per zone, where a tier is of constant step height and a zone is a group of tiers. The first zone covers the first half of the first Fresnel zone. The second zone covers the second half of the first Fresnel zone to the first half of the next Fresnel zone and all succeeding zones follow the pattern of the second zone. This is indicated in Figure 19. Also shown are the tier radii r_z. These are phase matched such that each tier covers the same phasor angle in the vibration curve equal to π/(tiers_per_zone) radians.

DIFFRACTION LOSSES

As the mirror radius increases, more higher order transverse modes begin to oscillate since the peak of the outermost ripple of each mode extends outward in the radial direction. To illustrate this Figure 20 shows the first three Laguerre-Gaussian TEM_{p0} transverse modes.

The mode half-width x_p is defined as the peak of the outermost ripple of the Laguerre-Gaussian pattern [18]. This half-width or spread is proportional to the radial index p in approximately the form

$$x_p \approx (p)^{1/2} w \; . \tag{74}$$

The number of transverse modes that will fit within the mirror radius r is given by the radial index p so that

$$x_p \le r \quad \text{or} \tag{75}$$

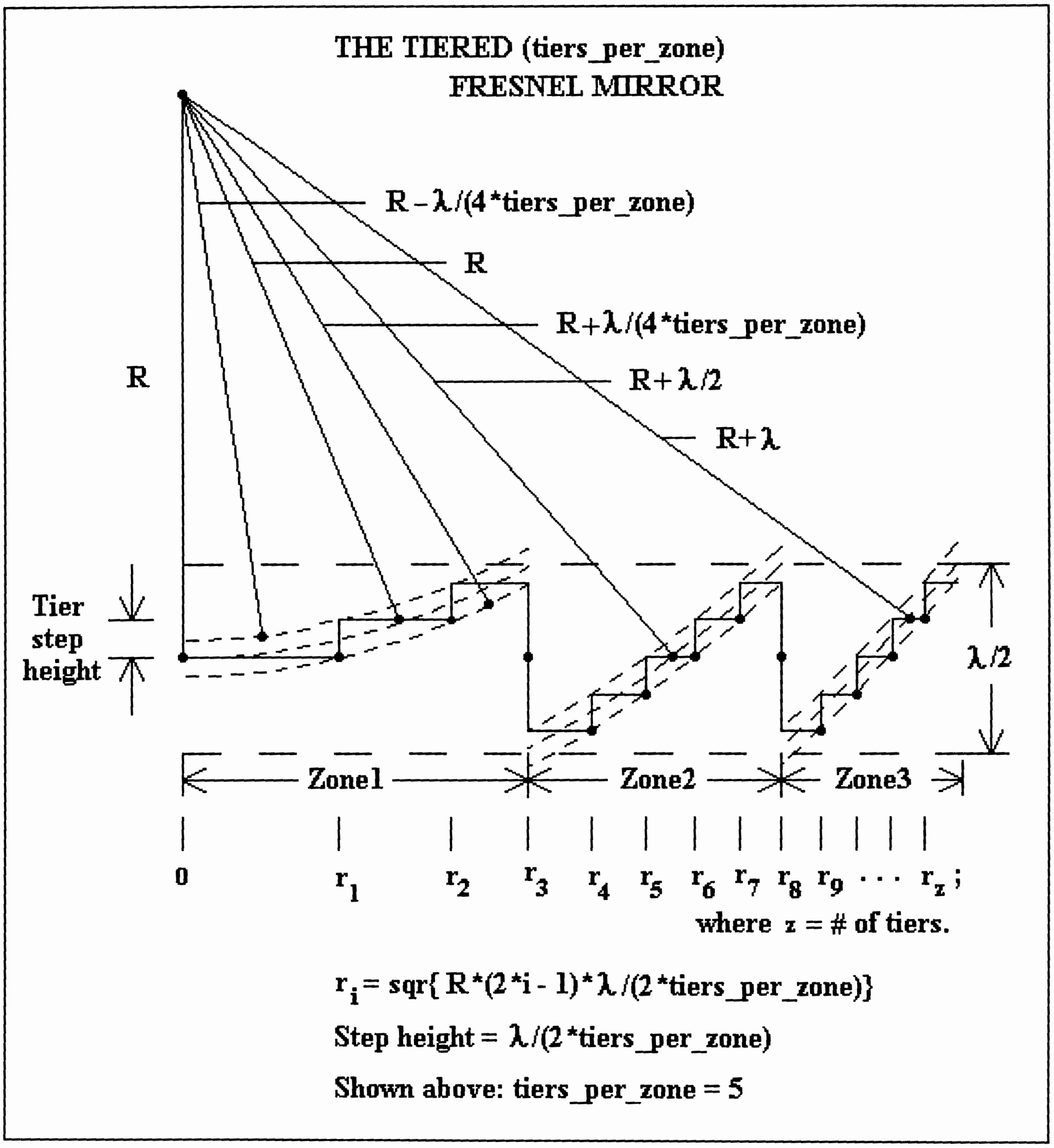


Figure 19. Critical points of the Tiered Fresnel Mirror. The dotted lines are the edge boundaries of the tiers. Each tier spans an equal phase.

$$w(p)^{1/2} \leq r \quad => \quad p \leq r^2/w^2 . \tag{76}$$

For the symmetric confocal resonator $w^2 = \lambda d/\pi$ and using the Fresnel Number N we have

$$p \leq r^2/w^2 \quad => \quad p \leq \pi N . \tag{77}$$

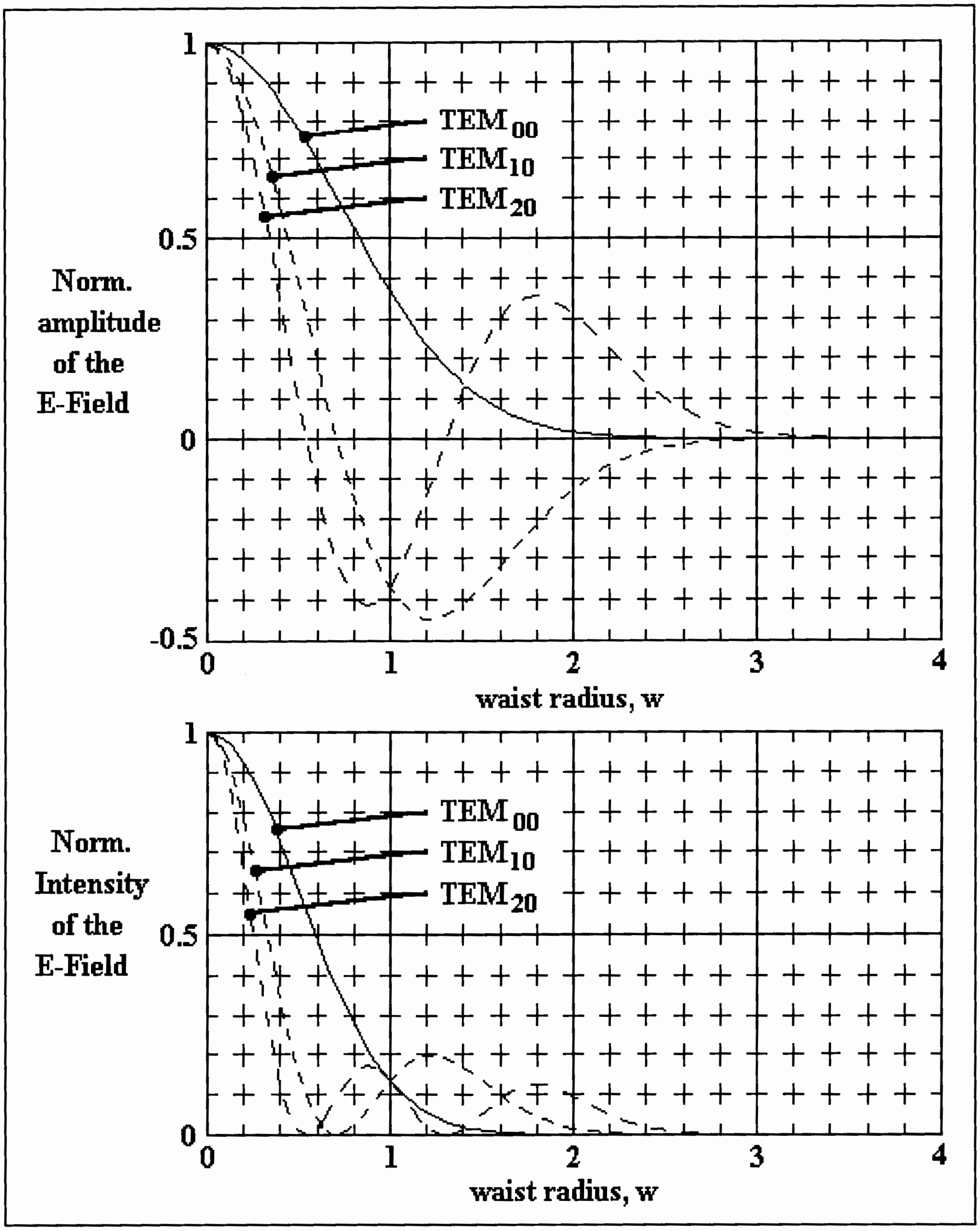


Figure 20. The first three Laguerre-Gaussian TEM_{p0} modes vs. the waist radius. The modes extend more radially with increasing mode index p.

For example, if the mirror radius equals the waist at the mirror then both TEM_{00} and TEM_{10} modes will fit within the mirror and higher modes will spill over the mirror's edge and will quickly die out due to diffraction. Figure 20 indicates, that when r equals w, the TEM_{10} mode is going to be attenuated much more rapidly than the TEM_{00} mode since a major portion of the TEM_{10} mode's spatial extent is truncated by the mirror's edge.

As the mirror radius increases, more TEM modes will oscillate in a low loss condition. Notice that the TEM_{00} mode at any mirror radius has the least loss per transit or pass of any of the TEM modes. Thus the term *dominant* or *fundamental mode* is commonly referred to the TEM_{00} mode. The term transit refers to the beam's traverse from one mirror to the other.

The Fox and Li Method

In 1961, when the laser was in its infancy, A.G. Fox and Tingye Li wrote a classical paper investigating diffraction effects in symmetric laser resonators[19]. They introduced a method to determine the steady state Electric Field or E-Field from an initially launched wave within the resonator. The method they used is still valid and used today. The Fox and Li method is used to study the diffraction loss in a variety of different mirrors such as a hole in the center or a mirror consisting of annular rings. Any arbitrary mirror shape can be analyzed as long as the physical dimensions of the mirrors are accurately known. This includes sensitivity studies where the effects of mirror imperfections can be modeled.

The method used is as follows: 1) an initial arbitrary wave is launched from one mirror, say M_2, towards the other mirror M_1; 2) the E-Field distribution at M_1; is computed by use of the Huygens-Fresnel diffraction integral evaluated at M_2; 3) the wave is then reflected from M_1 and the new E-Field at M_2 is re-computed in similar manner from the calculated E-Field at M_1; and 4) the computation of the E-Field distribution is repeated over and over again for subsequent successive transits of the transformed wave until a steady state is reached.

This iterative method is analogous to the physical process involved in the resonator when the laser beam is first initiated by noise or the spontaneous emission of the laser medium.

Symmetric Resonators. We now will apply the Fox and Li method to determine the steady state E-Field distribution in a symmetric resonator.

If the mirror radius is large compared to the wavelength, the E-Field is very nearly transverse in spatial extent, and the E-Field is uniformly polarized in one dimension then the scalar form of the Huygens-Fresnel diffraction integral can be used [20]. The E-Field due to the illuminated aperture A is given by

$$E_{qpm} = \frac{i}{2\lambda} \int_A E_{(q-1)pm} \frac{e^{-ikD}}{D} (1 + \cos(\theta))\, dS \,, \tag{78}$$

where k is the propagation constant of the medium; D is the distance from a point on the aperture to the point of observation; θ is the angle that D makes with the unit normal to the aperture; q is the number of transits that the beam makes; E_{0pm} is the initial wave launched from the aperture M_2; the aperture A is either M_1 if q is even or M_2 when q is odd; and pm are the radial and azimuthal Laguerre-Gaussian TEM mode indices.

After many q transits the initial E-Field will eventually reach a steady state. This is when the E-Fields at each mirror differ only by a complex constant. Thus we can write

$$E_{qpm} = (\gamma_{ST})_{pm}{}^{-q}\, v_{pm} = \exp[-q \ln(\gamma_{ST})_{pm}]\, v_{pm} \,, \tag{79}$$

where v_{pm} is a constant distribution function, $(\gamma_{ST})_{pm}$ is a complex constant independent of position coordinates and the ST denotes a single transit.

The logarithm of $(\gamma_{ST})_{pm}$ is the single transit propagation constant associated with the normal mode corresponding to a steady-state solution and specifies the attenuation and phase shift that the wave suffers during each transit.

When we substitute $(\gamma_{ST})_{pm}{}^{-q}\, v_{pm}$ for E_{qpm} and $(\gamma_{ST})_{pm}{}^{-(q-1)}\, v_{pm}$ for $E_{(q-1)pm}$ in the diffraction integral we obtain the integral equation

$$v_{pm} = (\gamma_{ST})_{pm} \int_A v_{pm} K_A \, dS_A \,, \tag{80}$$

where $K_A = (i/2\lambda D)(1 + \cos(\theta))e^{-ikD}$ = the kernel of the integral equation, and $(\gamma_{ST})_{pm}$ is the eigenvalue to the eigensolution v_{pm} of the integral equation. The distribution function v_{pm}, which satisfies the integral equation, is the normal mode of the symmetric resonator defined at the mirror surface.

The ratio of $E_{(q+1)pm}$ to E_{qpm} is less than one due to spill over of the beam at the mirror edge caused by diffraction. This is given by

$$\frac{E_{(q+1)pm}}{E_{qpm}} = \frac{(\gamma_{ST})_{pm}{}^{-(q+1)} v_{pm}}{(\gamma_{ST})_{pm}{}^{-q} v_{pm}} = (\gamma_{ST})_{pm}{}^{-1} \,. \tag{81}$$

The fractional power loss at the mirrors is given in Table I.

TABLE I

FRACTIONAL POWER LOSS IN A SYMMETRIC RESONATOR

Single Transit (ST)		Round-Trip (RT)
$E_{(q+1)pm} = (\gamma_{ST})_{pm}{}^{-1} E_{qpm}$		$E_{(q+2)pm} = (\gamma_{RT})_{pm}{}^{-1} E_{qpm}$
Fractional power loss:		Fractional power loss:
$1 - \lvert\gamma_{ST}\rvert_{pm}{}^{-2}$	$(\gamma_{ST})_{pm}{}^2 = (\gamma_{RT})_{pm}$	$1 - \lvert\gamma_{RT}\rvert_{pm}{}^{-2}$

Non-Symmetric Resonators. We will now apply the Fox and Li method to a non-symmetric resonator. A round-trip must be studied since no eigenvalues of the integral equation exist for the single pass. This is because the different mirrors cause the E-Fields at each mirror to be different in spatial extent during the steady state condition. We will end up with a double integral equation since integration is performed over each mirror.

This round-trip analysis can be considered as general since a symmetric resonator can be analyzed by using this technique.

We begin with the diffraction integral with an initial wave E_{0pm} propagating one transit from mirror M_2 in the form

$$E_{1pm} = \frac{i}{2\lambda} \int_{M_2} E_{0pm} \frac{e^{-ikD_2}}{D_2} (1 + \cos(\theta_2))\, dS_2 \,. \tag{82}$$

After a reflection at M_1, the beam completes a round-trip by making another transit and we obtain

$$E_{2pm} = \frac{i}{2\lambda} \int_{M_1} E_{1pm} \frac{e^{-ikD_1}}{D_1} (1 + \cos(\theta_1))\, dS_1 \,. \tag{83}$$

By substitution of E_1 into the equation for E_2 we have

$$E_{2pm} = \int_{M_1}\int_{M_2} E_{0pm}\, K_1\, K_2\, dS_2\, dS_1 \,, \tag{84}$$

where $K_j = (i/2\lambda D_j)(1 + \cos(\theta_j))e^{-ikD_j}$; $(j = 1,2)$ is the kernel of the double integral equation.

After many t round-trips a steady state evolves. The rate of convergence is a function of: 1) the form of the input wave E_{0pm}; and 2) the Fresnel number N. Again we describe the E-Field after t round-trips as

$$E_{tpm} = (\gamma_{RT})_{pm}^{-t}\, v_{pm} \,, \tag{85}$$

where the subscript RT denotes a round-trip and again v_{pm} is a constant distribution function and $(\gamma_{RT})_{pm}$ is a complex constant independent of position coordinates. The logarithm of $(\gamma_{RT})_{pm}$ is the round-trip propagation constant associated with the normal mode and specifies the attenuation and phase shift that the wave suffers during each round-trip.

When we substitute $(\gamma_{RT})_{pm}^{-t}\, v_{pm}$ for E_{2pm} and $(\gamma_{RT})_{pm}^{-(t-1)}\, v_{pm}$ for E_{0pm} in the diffraction integral, we obtain the double integral equation

$$v_{pm} = (\gamma_{RT})_{pm} \int_{M_1} \int_{M_2} v_{pm}\, K_1\, K_2\, dS_2\, dS_1 \,. \tag{86}$$

The ratio of the E-Fields at each mirror within a round-trip is meaningless since each has different spatial extents, while $E_{(t+1)pm}$ to E_{tpm} specifies the attenuated field given by

$$\frac{E_{(t+1)pm}}{E_{tpm}} = \frac{(\gamma_{RT})_{pm}{}^{-(t+1)}\, v_{pm}}{(\gamma_{RT})_{pm}{}^{-t}\, v_{pm}} = (\gamma_{RT})_{pm}{}^{-1} \,. \tag{87}$$

The fractional power loss per round-trip is given in Table II.

TABLE II

FRACTIONAL POWER LOSS IN A NON-SYMMETRIC RESONATOR

Single Transit (ST)		Round-Trip (RT)
$E_{(t+½)pm} \neq (\gamma_{RT})_{pm}{}^{-½}\, E_{tpm}$		$E_{(t+1)pm} = (\gamma_{RT})_{pm}{}^{-1}\, E_{tpm}$
Fractional power loss:		Fractional power loss:
undefined	$(\gamma_{ST})_{pm}$ is undefined	$1 - \lvert\gamma_{RT}\rvert_{pm}{}^{-2}$

To be thorough when discussing the losses in a resonator we must include all Laguerre-Gaussian TEM modes. Any transverse wave can be decomposed into a complete set of modes.

For example, the input wave specified by E_0 is actually a composite of Laguerre-Gaussian modes, that by superposition, comprise E_0 as

$$E_0(r,\varphi) = \sum_{pm} c_{pm}\, E_{pm}(r,\varphi) \,. \tag{88}$$

Each transverse mode in the integral equation has its own eigenvalue: $(\gamma_{ST})_{pm}$ for a symmetrical resonator; and $(\gamma_{RT})_{pm}$ for a non-symmetric resonator.

In a symmetrical resonator, when a steady state is achieved after q transits, the E-Field at the mirror can be written as

$$E^{[q]}(r,\varphi) = \sum_{pm} c_{pm} \, (\gamma_{ST})_{pm}^{-q} \, E_{pm}(r,\varphi) \, . \tag{89}$$

The relative amplitude of each transverse mode after q transits will in general be different and will be exponentially attenuated by $|\gamma_{ST}|_{pm}^{q}$. The TEM_{00} mode will have the largest eigenvalue or lowest loss per transit. All other modes have smaller eigenvalues and thus will die out quicker at different rates depending on the value of their particular eigenvalue.

To regress, we rewrite the integral equations for a symmetric and a non-symmetric resonator as

$$v_{pm} = (\gamma_{ST})_{pm} \int_A v_{pm} \, K_A \, dS_A \tag{90}$$

and

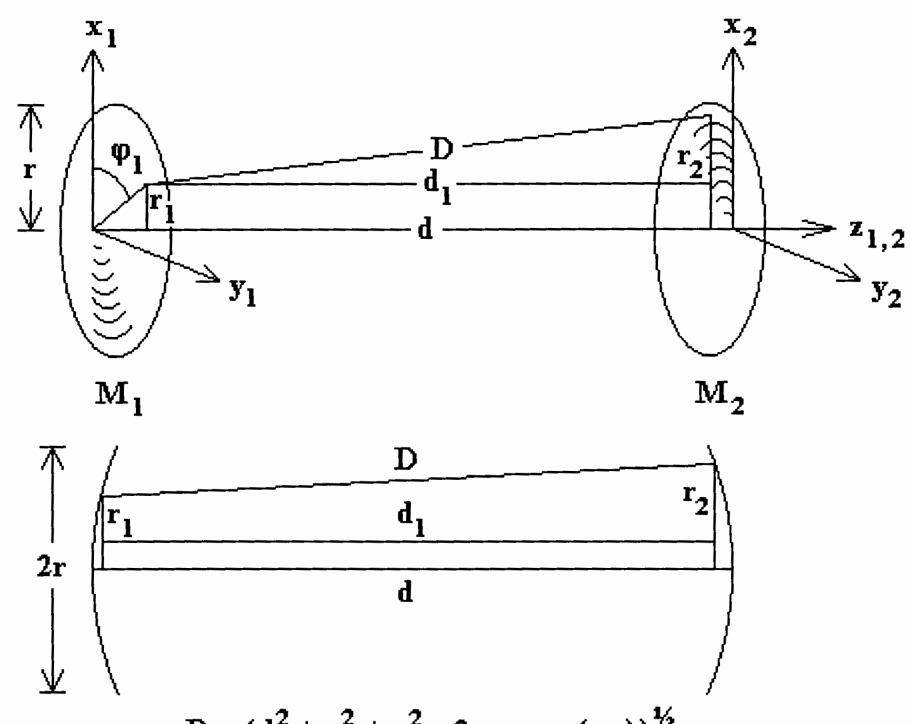
$$v_{pm} = (\gamma_{RT})_{pm} \int_{M_1} \int_{M_2} v_{pm} \, K_1 \, K_2 \, dS_2 \, dS_1 \tag{91}$$

respectively.

CHAPTER IV

COMPUTER SIMULATIONS

For the study of Tiered Fresnel Mirrors, we will use only circular symmetric mirrors and a Laguerre-Gaussian beam where the azimuthal index m equals zero. We will work in cylindrical coordinates with the set-up shown in Figure 21.



$$D = (d_1^2 + r_1^2 + r_2^2 - 2\,r_1 r_2 \cos(\varphi_1))^{1/2}$$

$$\varphi_2 \equiv 0\,;\;\; 0 \le \varphi_1 \le 2\pi\,;\;\; 0 \le r_{1,2} \le r$$

Figure 21. Geometry of a Symmetric Resonator.

The Fox and Li method is used where, at the observation mirror, a radial line at $\varphi = 0$ is taken and the E-Field is determined at equal incremented points from the mirror center to its edge along this radial line. Since cylindrical symmetry exists, the field along this radial line can be rotated from 0 to 2π in equal increments, and duplicated to fill the observation mirror with the observed E-Field distribution.

This process is replicated at the other mirror due to the beam's reflection and iterated back and forth until the E-Field varies negligibly in three consecutive transits.

We begin with the Huygens-Fresnel diffraction integral where once again stated is

$$E_{qpm} = \frac{i}{2\lambda} \int_A E_{(q-1)pm} \frac{e^{-ikD}}{D} (1 + \cos(\theta))\, dS\ , \tag{92}$$

This integral can be separated into two integrals since both E_{qpm} and $E_{(q-1)pm}$ have real and imaginary components.

We can set $\cos(\theta) = d_1 / D$ since the mirror sizes are small compared to the spacing d (d_1 is shown in Figure 21). We will also replace the following: E_{qpm} by $(E_{qpm})_R + i(E_{qpm})_I$; $E_{(q-1)pm}$ by $(E_{(q-1)pm})_R + i(E_{(q-1)pm})_I$; e^{-ikD} by $[\cos(kD) - i\sin(kD)]$; and dS by $\rho d\rho d\varphi$. After some manipulation we obtain the real and imaginary parts of the E-Field as

$$(E_{qpm})_R = \int_0^{2\pi} \frac{1}{2\lambda} \int_0^{r} \frac{(1+d_1/D)}{D} [(E_{(q-1)pm})_R \cos(kD) - (E_{(q-1)pm})_I \sin(kD)]\, \rho d\rho d\varphi \tag{93}$$

and

$$(E_{qpm})_I = \int_0^{2\pi} \frac{1}{2\lambda} \int_0^{r} \frac{(1+d_1/D)}{D} [(E_{(q-1)pm})_R \sin(kD) + (E_{(q-1)pm})_I \cos(kD)]\, \rho d\rho d\varphi\ . \tag{94}$$

These two equations are iterated with each transit that the beam makes until a steady state solution of the E-Field distribution is found.

A computer program was written to determine the loss per pass (or performance) of various resonator types and in particular to compare the performance of the novel Tiered Fresnel Mirror to that of the common Spherical Mirror. The program is listed in Appendix A along with a cross reference mapping of all the variables used which is found in Appendix B.

An example output of a short computer run having only five round-trips is shown in Figure 22. Notice that the input wave is a plane wave and already it has been drastically transformed from a horizontal line equal to 1 for the normalized amplitude to that of the solid curve in sections U and V in Figure 22.

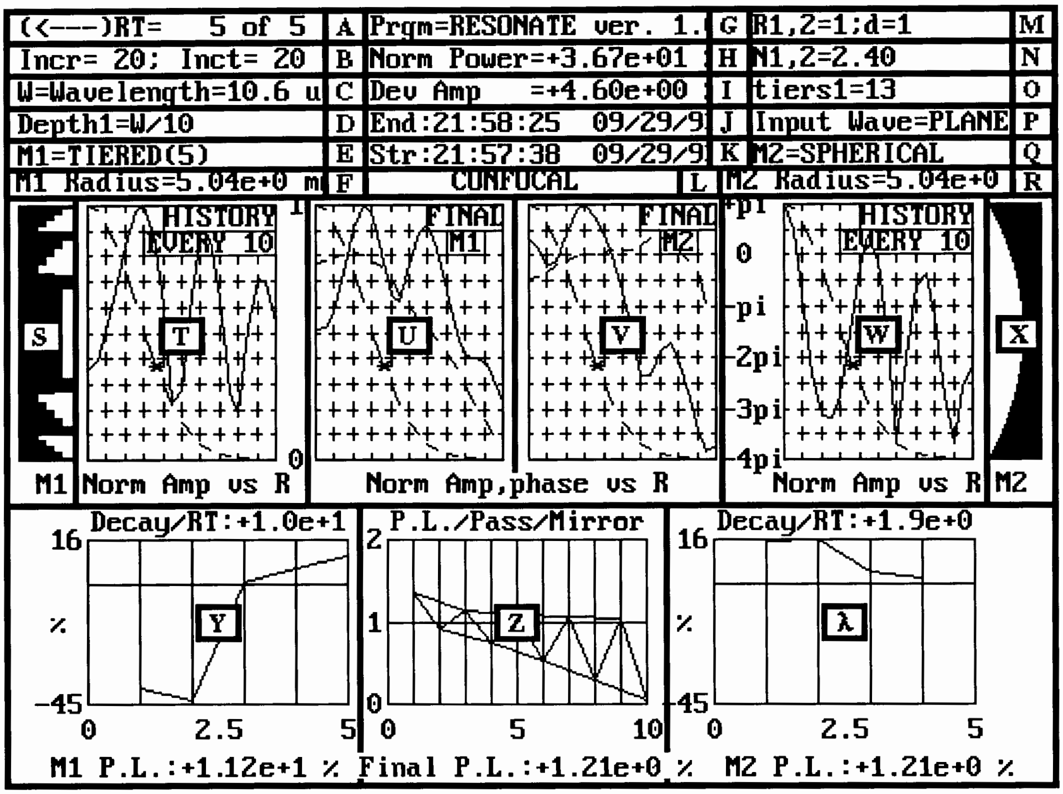


Figure 22. An example output of a computer simulation of a Tiered(5) Fresnel Mirror paired with a Spherical Mirror.

Each section of the figure is labeled with a letter A-Z and λ. The explanation of each section is as follows:

A) The arrow indicates the direction the beam travels. "RT" stands for round-trips.

B) "Incr" is the number of radial increments and "Inct" is the number of azimuthal increments used in the integration of the mirror.

C) "W=Wavelength" is the wavelength of the oscillating beam.

D) "Depth1,2" indicates the etch depth or step height of the tier on mirrors M_1 and M_2. Displayed only for Tiered Fresnel Mirrors.

E) "M1" is the mirror on the left. The available mirror types for M_1 and M_2 are Spherical, Parabolic, Plane, Tiered and Fresnel. For the Spherical type one can use either positive or negative mirror curvatures. For the Tiered Fresnel Mirror one can use any odd number of tiers per zone. This number is denoted within the parentheses.

F) "M1 Radius" is the radial dimension of mirror M_1.

G) "Prgm = RESONATE ver 1.0" is the program name and current version.

H) "Norm Power" is the current power illuminated on the proper mirror relative to the power of the input beam. This value decreases with each reflection of the beam due to beam spill over at the mirror's edge.

I) "Dev Amp" is the deviation or change in area under the amplitude curve relative to the last calculated area of the particular mirror. In other words the value 1 - (dev_amp_q / dev_amp_{q-1}), where q is the transit number. This value is evaluated at M_1 if q is even or at M_2 if q is odd. If three consecutive round-trips of the beam all yield dev_amp values of less than 7e-4 % then the program terminates and it is assumed that a steady state has been reached.

NOTE: The program actually plots the square root of the intensity yielding curves in the positive domain. Hereafter use of the term amplitude pertaining to the program will actually be the square root of the intensity.

J) "End" indicates the time that the program stops.

K) "Str" indicates the time that the program starts.

L) "CONFOCAL" is the special resonator type. The types displayed are dependent upon the system parameters. The available resonator types are Plane-Parallel, Half Confocal, Half Concentric, Confocal, Concentric, and for non-special resonators; Confined Beam or Unconfined Beam.

M) "R1,2" indicates the radius of mirror curvature of mirrors M_1 and M_2. This value is variable for the Spherical Mirror and fixed at one meter for both Tiered and Fresnel Mirrors. If the mirror is Parabolic the value displayed is PARAB or if the mirror is Plane then PLANE is displayed. "d" is the cavity length or mirror separation.

N) "N1,2" is the Fresnel number of mirrors M_1 and M_2. This is the number of Fresnel zones that span a mirror as seen from the center of the other mirror.

O) "tiers1,2" is the total number of tiers of mirrors M_1 and M_2. Displayed only for Tiered Fresnel Mirrors.

P) "Input Wave" is the form of the initial wave. A plane or Gaussian wave are the available choices.

Q) "M2" is the mirror type shown on the right side.

R) "M2 Radius" is the radial dimension of mirror M_2.

S) This is an illustration of mirror M_1.

T) This graph is a history of mirror M_1's normalized amplitude vs. the mirror radius. Every 10th round-trip the E-Field distribution is plotted, i.e. 1, 11,21,31, ··· round-trips are plotted. In this section (as well as sections U, V, and W) the following pertains: 1) the horizontal axis is the mirror radius where the center (r = 0) is located at the left and the mirror's edge ($r = r_{max}$) is on the right; 2) the vertical axis is the normalized amplitude with the value of one at the top and zero at the bottom; 3) a dashed Gaussian curve is the theoretical E-Field distribution of the Laguerre-Gaussian TEM_{00} mode independent of the mirror radius; and 4) an asterisk on the dotted Gaussian curve that indicates the $1/e$ point which is the waist radius at the mirror.

U) This graph is mirror M_1's final E-Field distribution (the solid curve) and a plot of the phase of the final E-Field both vs. the mirror radius. In this section (as well as in section V) the following pertains: 1) the value of the phase at the top is $+\pi$ whereas at the bottom the phase value is -4π; and 2) the phase curve (the dashed curve other than the dashed Gaussian curve) is relative to the maximum value of the normalized amplitude of the E-Field distribution where at this point the phase is defined as zero radians. A positive phase value indicates a leading value of the phase of the wave front and a negative phase value indicates a lag in phase with respect to the phase of the maximum value of the normalized amplitude.

V) This graph applies to mirror M_2. Refer to section U's explanation.

W) This graph applies to mirror M_2. The explanation of this section is similar to section T.

X) This is an illustration of mirror M_2.

Y) This graph is mirror M_1's convergence of decay per round-trip vs. round-trip. The decay/RT, the vertical axis, is a linear scale. The graph is a measure of the change in M_1's area under the E-Field distribution relative to the last area's value. The horizontal line in the graph show the zero level.

When the decay/RT is plotted at the zero level, no change in the area under the normalized amplitude curve has occurred in two consecutive round-trips at mirror M_1. At the bottom of the section is M_1's power loss relative to the energy reflected from mirror M_2.

Z) This is a graph of the percent power loss P.L. per pass per mirror vs. pass or transit. The P.L./Pass/Mirror, the vertical axis, is a log to the base 10 scale. For example: A value of two, the maximum attainable value, represents a 100% power loss and a value of zero represents a 1% power loss per pass. Three curves are plotted, one is a zigzag and the others are envelopes. One of the envelope curves begins at the first transit, this is the power loss due to mirror M_1. The other envelope curve begins at the second transit and is M_2's power loss curve. The zigzag curve just connects the two envelope curves and indicates sequential power loss of the system of two mirrors. At the bottom of the section is the either M_1's or M_2's power loss relative to the energy reflected from the other mirror depending upon which transit the beam is on.

λ) This is M_2's graph of the decay/RT vs. round-trip. The explanation of this section is similar to section Y.

We begin the computer simulations by investigating the effect of mirror radius with respect to the theoretical waist at the mirrors of a symmetric confocal resonator. The theory described in Chapter II *does not* take the mirror radius into account.

First, let us determine the power in a transverse infinite plane that is carried by the fundamental mode of a Gaussian beam. Then, find the power contained in a transverse circular plane of finite radius equalling the mirror radius.

The power P is defined as

$$P = \int I(x,y,z)\, da \,, \tag{95}$$

where $I(x,y,z) = |E(x,y,z)|^2$ = the optical intensity and da is an incremental area. This can be rewritten as

$$P = \int I(\rho,z)\, da = \int I_0 \left[\frac{w_0}{w(z)}\right]^2 \exp\left[-\frac{2\rho^2}{w^2(z)}\right] da\,, \tag{96}$$

where $I_0 = |E_0|^2$ and $\rho^2 = x^2 + y^2$.

The power carried by the beam of infinite extent in the transverse direction is

$$P_{\rho=\infty} = I_0 \left[\frac{w_0}{w(z)}\right]^2 \int_0^\infty \exp\left[-\frac{2\rho^2}{w^2(z)}\right] \rho d\rho \int_0^{2\pi} d\varphi\,, \tag{97}$$

which reduces to

$$P_{\rho=\infty} = ½\, I_0\, (\pi w_0^2)\,. \tag{98}$$

Thus the power is ½ the peak intensity times the beam area.

Now we will determine the power contained within the finite circular area of radius equal to the mirror radius r. We have

$$P_{\rho=r} = I_0 \left[\frac{w_0}{w(z)}\right]^2 \int_0^r \exp\left[-\frac{2\rho^2}{w^2(z)}\right] \rho d\rho \int_0^{2\pi} d\varphi\,, \tag{99}$$

which reduces to

$$P_{\rho=r} = ½\, I_0\, (\pi w_0^2)\left[1 - \exp\left[-\frac{2r^2}{w^2(z)}\right]\right]. \tag{100}$$

The percent of the total power P_T carried within a circle of radius r is given by

$$\%P_T = \frac{P_{\rho=r}}{P_{\rho=\infty}} \times 100\% = \left[1 - \exp\left[-\frac{2r^2}{w^2(z)}\right]\right] \times 100\%\,. \tag{101}$$

For example, the power contained inside a circle of radius r=w(z) is about 86% of the total power. In other words, about 16% of the total power spills over the mirror's edge and is lost in the open unwalled resonator.

The beam radius at the mirrors of a Symmetric Confocal Resonator is

$w = (\lambda d/\pi)^{1/2}$. Since the Fresnel number N equals $r^2/\lambda d$, we can say $N = r^2/\pi w^2$. The loss in the Symmetric Confocal Resonator is governed by this Fresnel number N; a higher value of N means a smaller loss.

Figure 23 shows a plot of the difference between the theoretical and simulated power that spills over the mirror's edge in a Symmetric Confocal Resonator (d/R = 1) as well as a Symmetric Resonator where d/R = 1.5. The percent theoretical power loss is $\%P_T$ as is defined above (the value of the mirror radius r is found from a given N value where $r = (N\lambda d)^{1/2}$ and the value of the waist at the mirrors $w_{1,2}$ in a Symmetric Resonator is given by $w_{1,2} = (\lambda d/2\pi n)^{1/2}[2R^2(d(R - d/2))]^{1/4}$). The simulated power loss is obtained

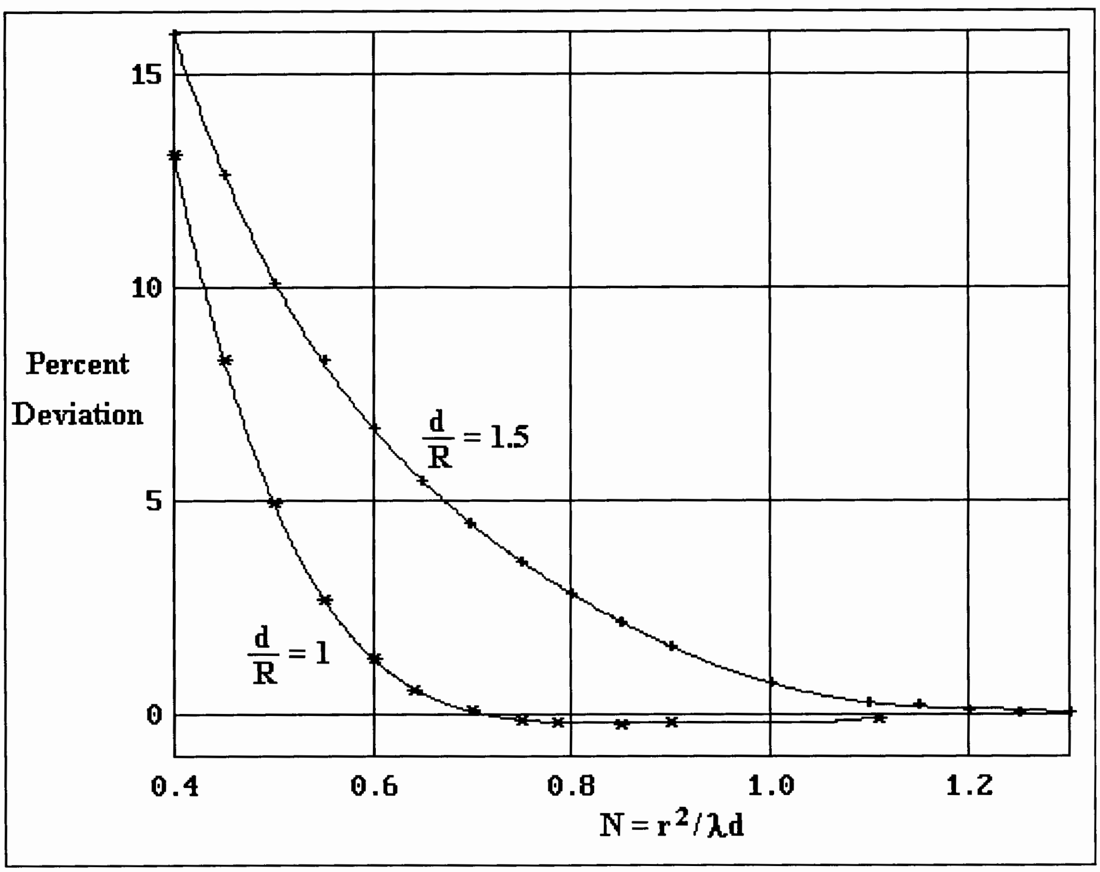


Figure 23. The difference between the theoretical and simulated values of loss per transit vs. the Fresnel number of a Symmetric Confocal Resonator (d/R=1) and another Symmetric Resonator where d/R=1.5.

from a computer run for a particular value of N. The Percent Deviation axis in the figure is defined as the difference between the simulated and theoretical power losses. The figure indicates that as N increases the simulated loss per transit approaches the theoretical value. This is because the theory assumes an effective mirror radius equal to infinity.

A trend-plot of four computer runs of a Symmetric Confocal Resonator where the mirror radius is varied from r=w to r=2w is shown in Figures 24 and 25. Notice the quick convergence of the E-Field to the theoretical dotted Gaussian profile in the number of round-trips. In the bottom of Figure 25 convergence to this Gaussian profile is not even close. The higher transverse modes are still in competition for oscillation even though 300 round-trips have transpired. In this instance 500 to 600 round-trips may be required such that the steady state profile is reasonably close to the theoretical Gaussian profile.

Another trend-plot of the percent completion of a Symmetric Resonator where d/R=1.5 is shown in Figures 26 and 27. The percent completion is varied from 25% to 100%. Actually the steady state has not quite been reached as indicated in the bottom of Figure 27 by the dev_amp value of -1.85e-01% as well as the decay graphs not quite reaching zero percent. This is close enough though for comparison purposes to call this 100% complete after 200 round-trips where 250 round-trips may be required for full convergence. The important issue of this trend plot is the profile of the beam that is shown in the middle graphs (sections U and V) of the four computer simulations.

A comparison between four Symmetric Confocal Resonators is shown in Figures 28 and 29 where the mirror type is varied. Figure 28 shows there is no difference in the loss per transit between Spherical and Fresnel Mirror types. Figure 29 shows that as the number of tiers per zone of the Tiered Fresnel Mirror is increased the loss per transit is decreased. Remember, as the limit of the number of tiers per zone approaches infinity the Tiered Fresnel Mirror becomes the Fresnel Mirror.

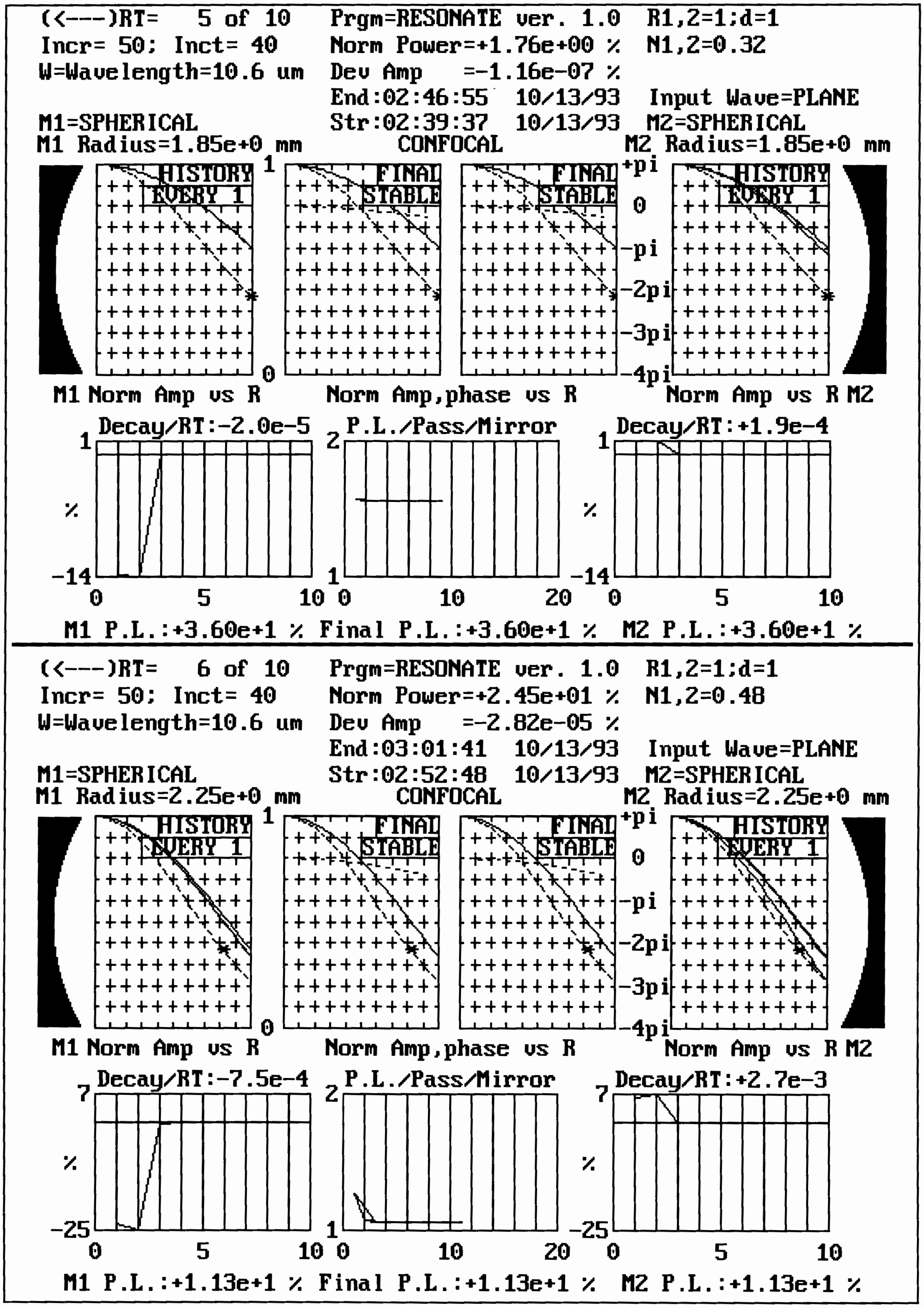


Figure 24. The mirror radius varies in a Symmetric Confocal Resonator in Figures 24 and 25. The top has r = w and the bottom has r = 1.2w.

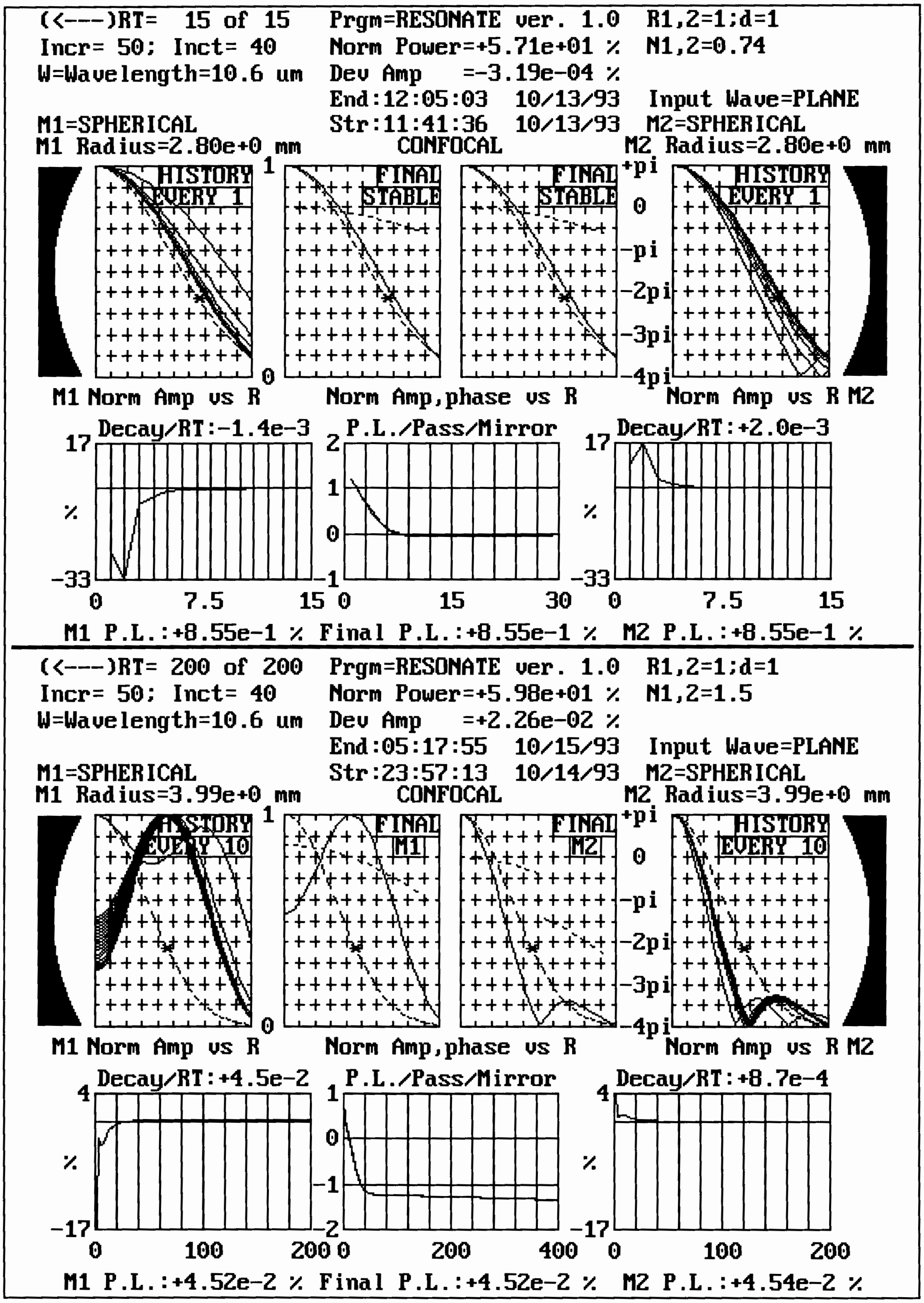


Figure 25. The mirror radius varies in a Symmetric Confocal Resonator in Figures 24 and 25. The top has r = 1.5w and the bottom has r = 2w.

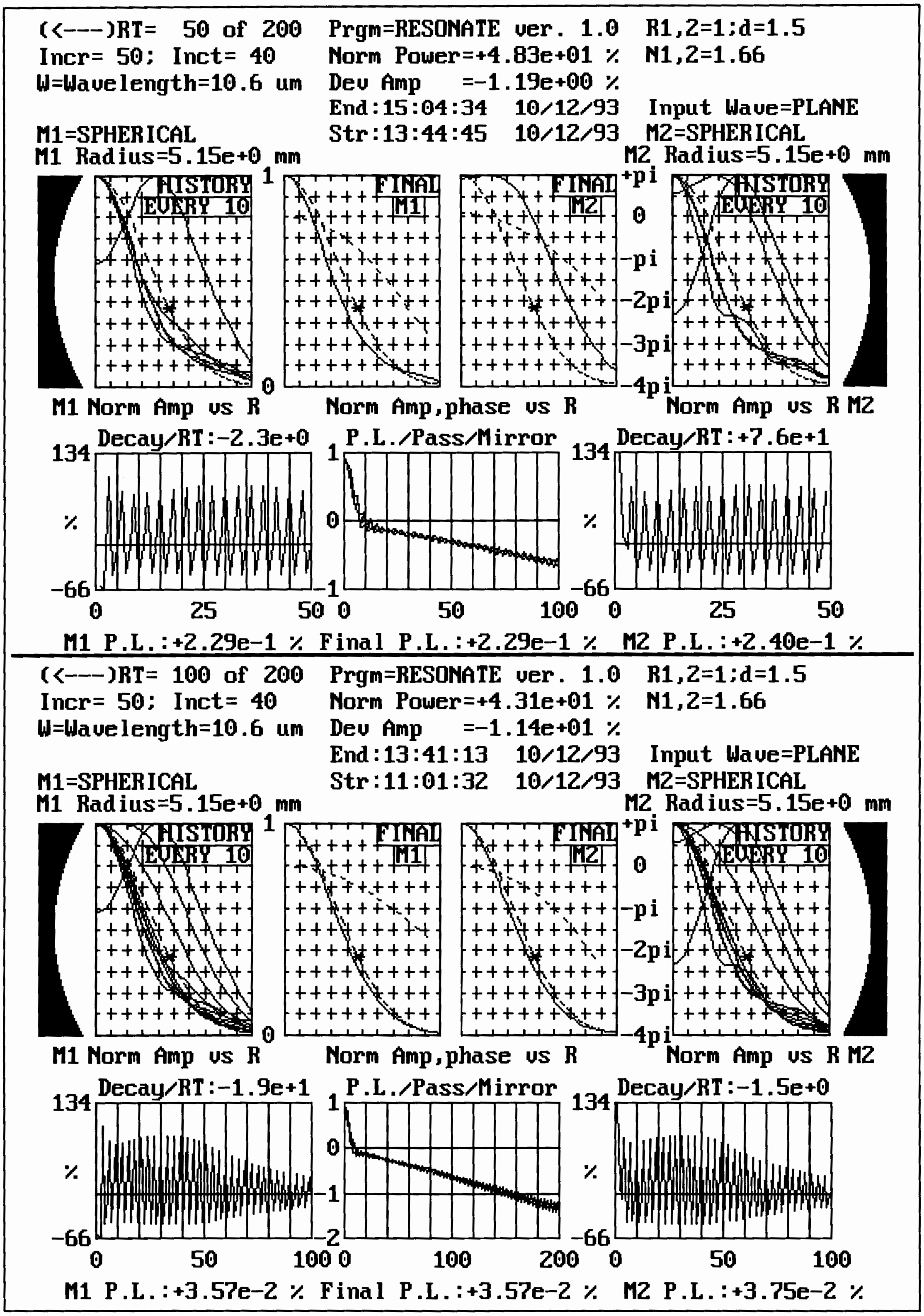


Figure 26. A 25% (top) and a 50% (bottom) completed run of a Symmetric Resonator (d/R=1.5). Compare with Figure 27.

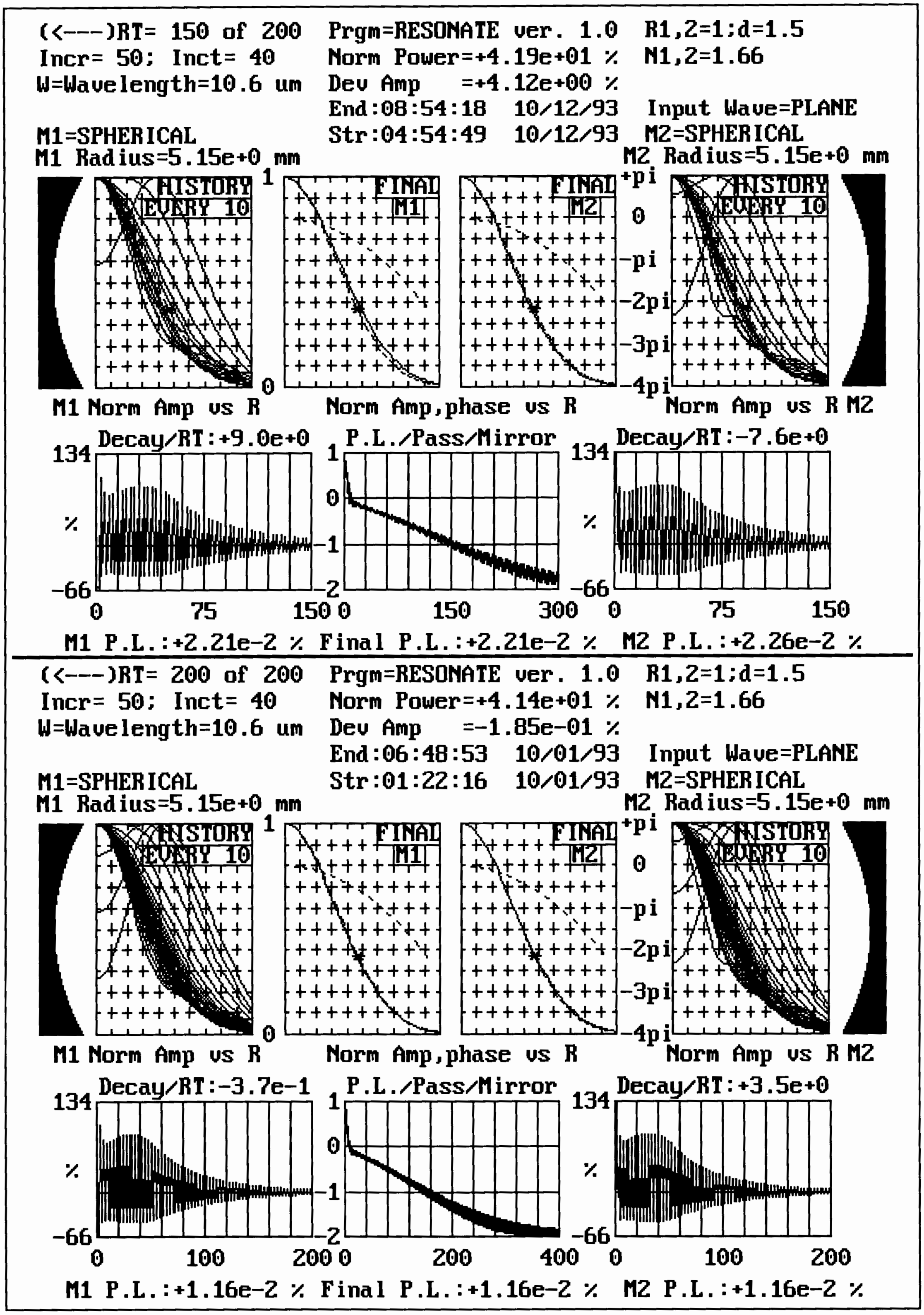


Figure 27. A 75% (top) and a 100% (bottom) completed run of a Symmetric Resonator (d/R=1.5). Compare with Figure 26.

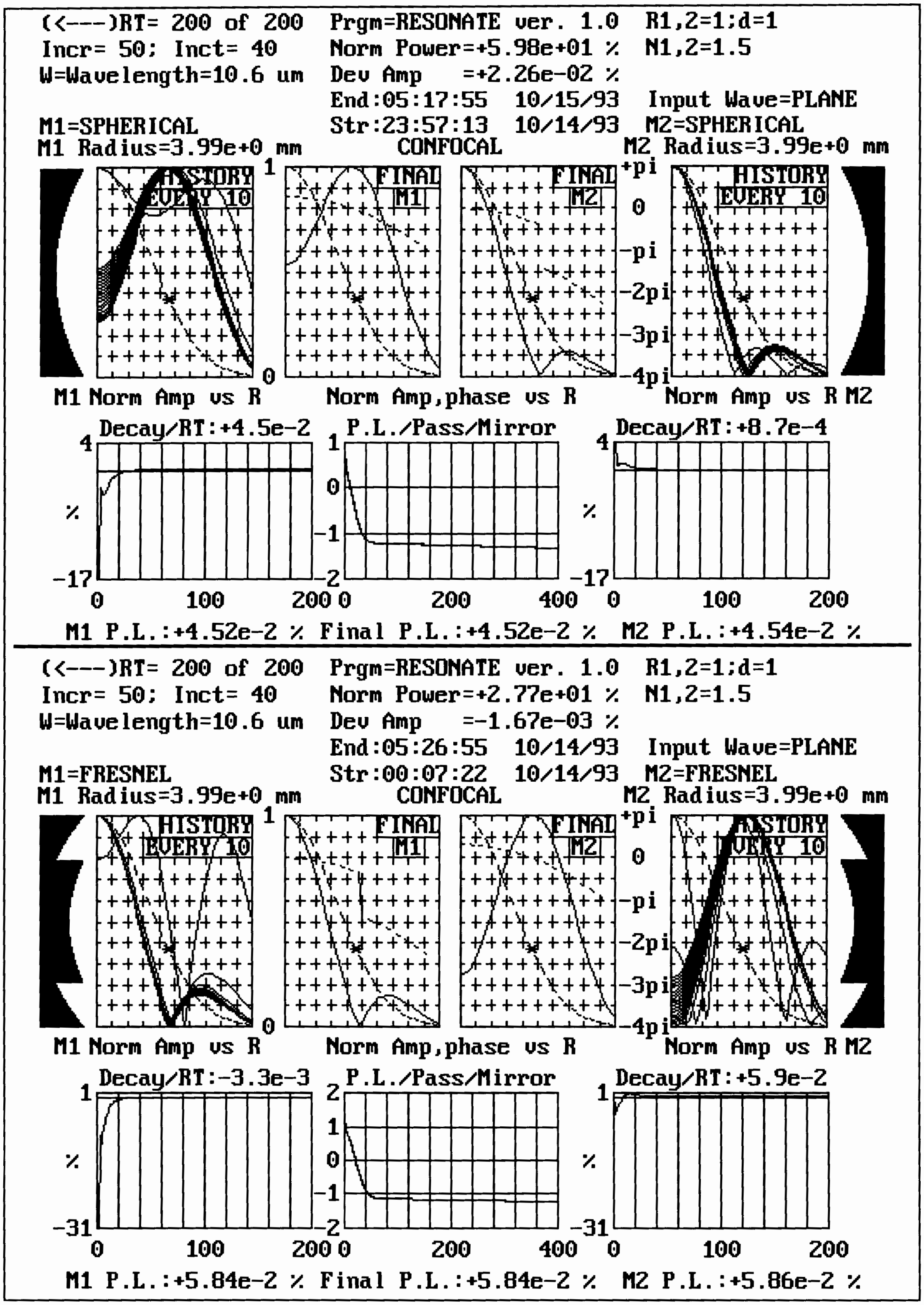


Figure 28. A comparison of mirror types of a Symmetric Confocal Resonator. Shown is a Spherical type (top) and a Fresnel type (bottom).

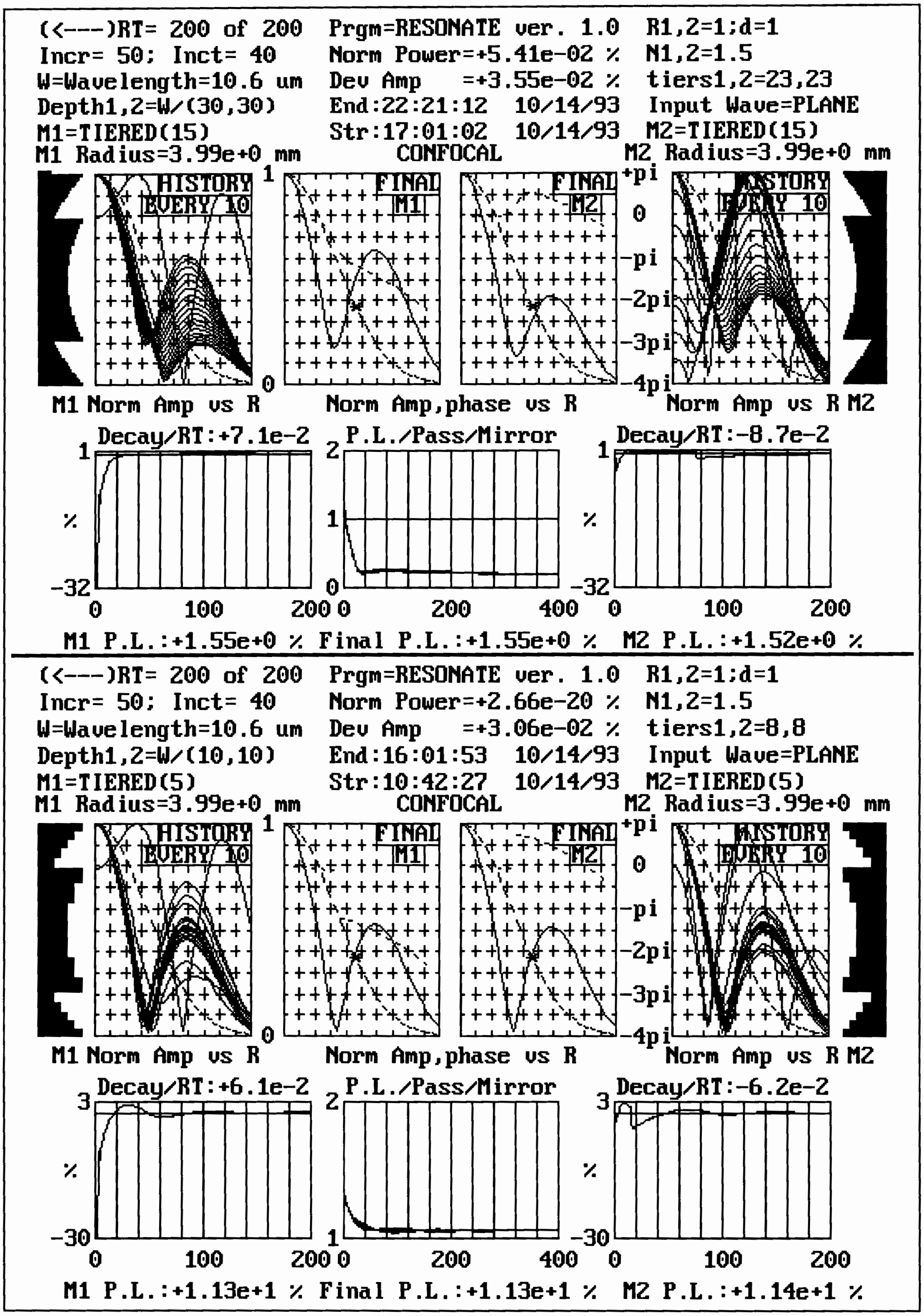


Figure 29. A Tiered(15) Fresnel (top) and a Tiered(5) Fresnel (bottom) Mirror types of a Symmetric Confocal Resonator. See Figure 28.

Another comparison between these four mirror types is shown in Figures 30 and 31 where a symmetric resonator of d/R=1.5 is used. Notice that for a given Fresnel number both the waist at the mirrors and the loss per transit are larger than for the symmetric confocal resonator. The diffraction loss can be readily seen from Figure 32.

Figure 32 is a paramount figure because it describes the feasibility of the tiered mirror's performance. It is shown that a Fresnel Mirror having more tiers per zone will have a better performance. This is because, when going from a more planar structure to that of a more curved one, better confinement of the beam occurs. Note also that the Fresnel Mirror and the Spherical Mirror has virtually identical loss profiles. This is due to both of the mirror's surfaces are constant phase surfaces. Also note that the leveling off of the power loss for the Tiered Fresnel Mirrors as the Fresnel Number increases is a possible artifact of the the resolution of the integration performed over the mirror surfaces. It is conjectured that as the number of increments is increased a more of a gradual leveling would occur.

A designer of a laser resonator system has to know the diffraction loss of the mirrors (found from Figure 32) as well as the gain of the laser medium. A high loss mirror teamed with a low gain laser medium is not a good match but rather just the opposite. For example, a tiered(5) Fresnel mirror (a tiered mirror with five zones per tier) in a d/R=1.5 symmetric resonator has a diffraction loss per transit of approximately 14% for a Fresnel number of one (see bottom of Figure 32). When teamed with a HeNe lasing medium of approximately 2% gain per transit it will not operate because amplification of the light is impossible due to a higher loss than gain. However, when a tiered(15) Fresnel mirror in a symmetric resonator of d/R=1.5 is teamed with a CO_2 laser medium, operation is now possible. This is because the diffraction loss per transit is approximately 1.5% while the CO_2 laser medium is approximately 10% gain per transit.

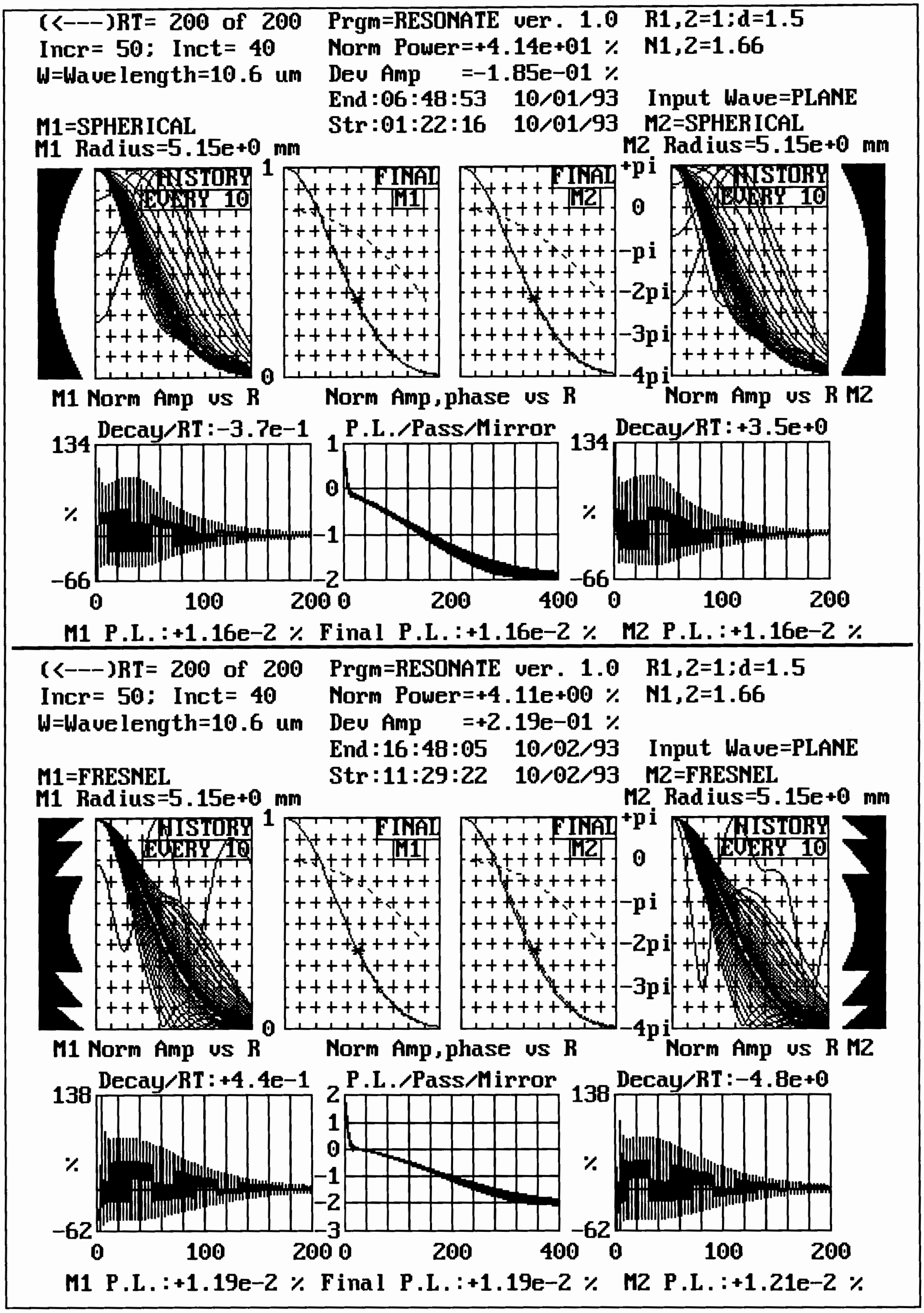


Figure 30. A comparison of mirror types of a d/R=1.5 Symmetric Resonator. Shown is a Spherical type (top) and a Fresnel type (bottom).

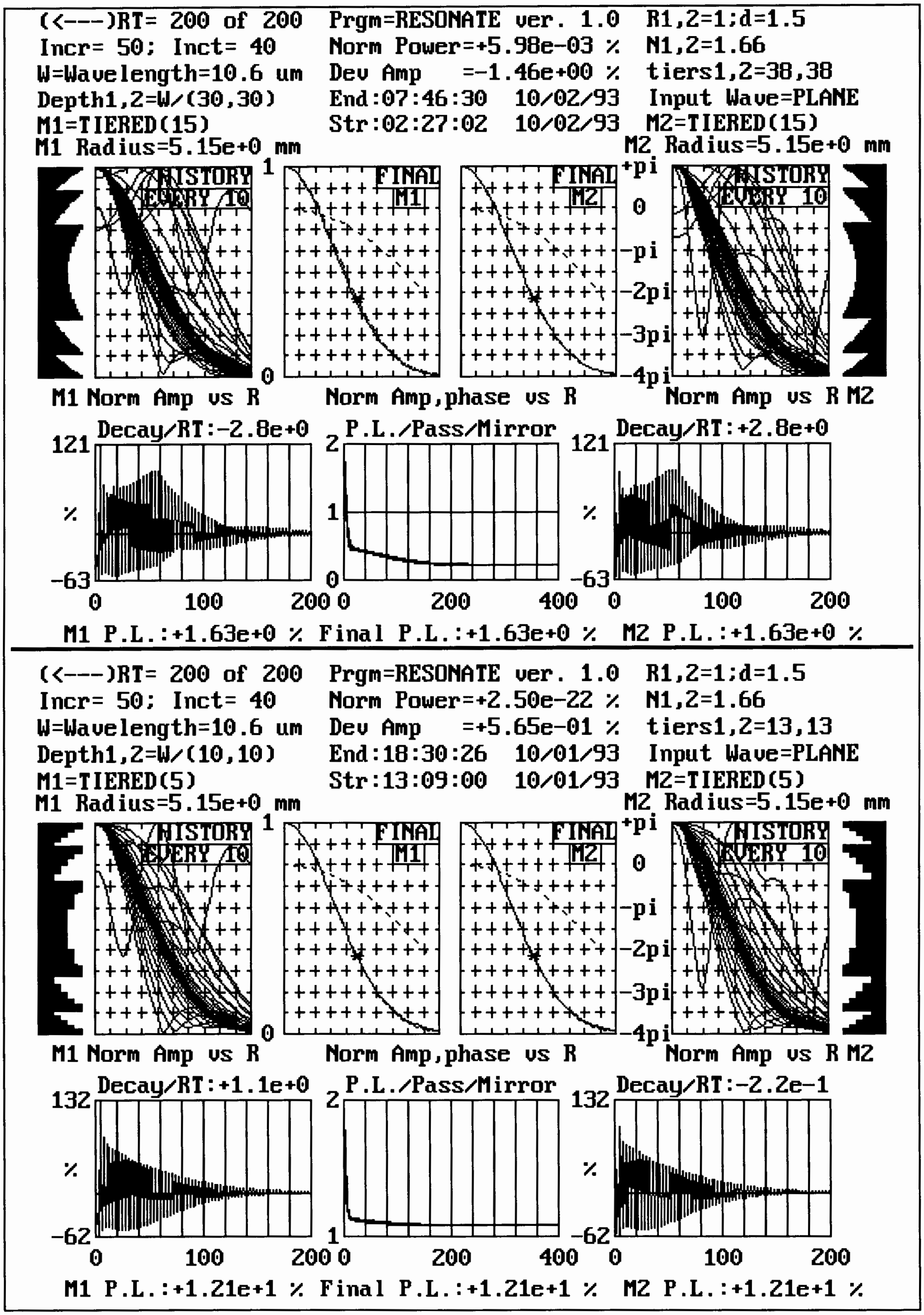
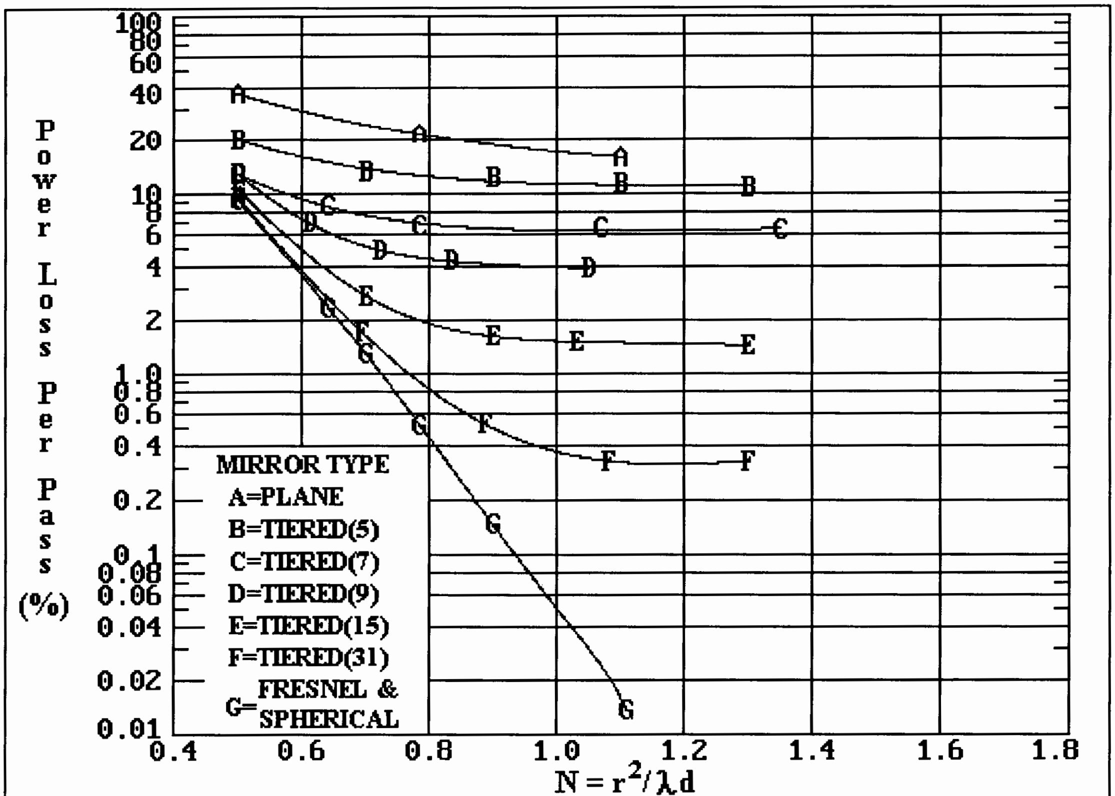


Figure 31. A Tiered(15) Fresnel (top) and a Tiered(5) Fresnel (bottom) Mirror types of a d/R=1.5 Symmetric Resonator. See Figure 30.



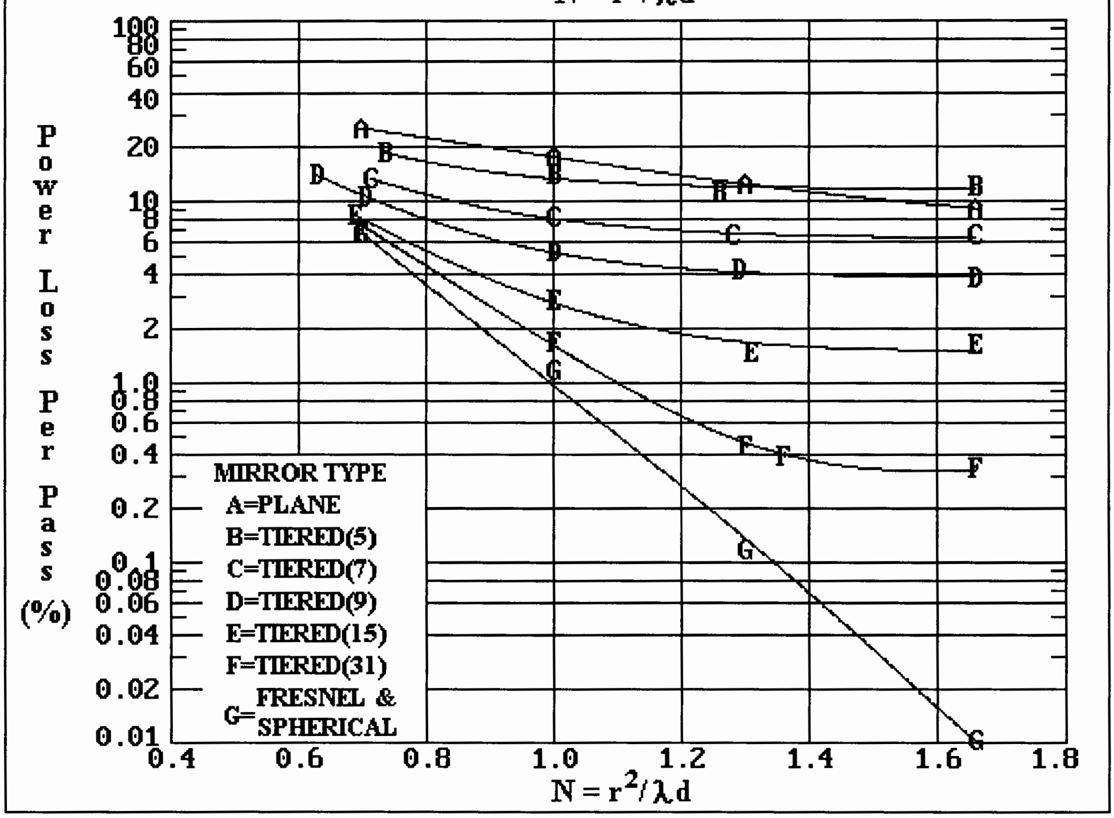


Figure 32. Mirror performances of a d/R=1 and a d/R=1.5 Symmetric Resonator.

CHAPTER V

INTEGRATED CIRCUITS PROCESSING

Tiered Fresnel Mirrors can be made efficiently in quantity by using segments of the Integrated Circuits Process. These segments consist of : 1) Photolithography; 2) Plasma Etching and 3) Metal Deposition. Also Low Pressure Chemical Vapor Deposition, LPCVD, oxide and LPCVD nitride may be used.

Two types of Tiered Fresnel Mirrors can be manufactured. One type is a simpler reflective mirror whereas the other type is a more involved transmittant mirror. The reflective type may be best used with a plane mirror. This configuration is the Half-Symmetric Resonator. The reflective mirror is made by sequential patterning and etching to form the tiers. Then a metal deposition is performed to form the fully reflective surface of the mirror. This mirror configuration is shown in Figure 33.

The other transmittant mirror type can be made by multiple depositions of LPCVD oxide and LPCVD nitride, patterning, etching and a metal deposition for the partially transmittant mirror surface. A cross-sectional view is shown in Figure 34. A zoom of this mirror is shown in Figure 35. The wavefront of the beam is concave curved when arriving at the mirror surface. What transmits through the metallic film exits through the mirror with a wavefront that is parallel to the back side of the mirror surface. This can be seen by observing that the sinusoidal curves, representing the E-Field distribution of the beam, are all in the same phase at the back side of the mirror. The amplitude of the sinusoidal curves is the same to illustrate the phase retardation when travelling through the mirror whereas the amplitude is exponentially diminished in the radial dimension as shown in Figure 34.

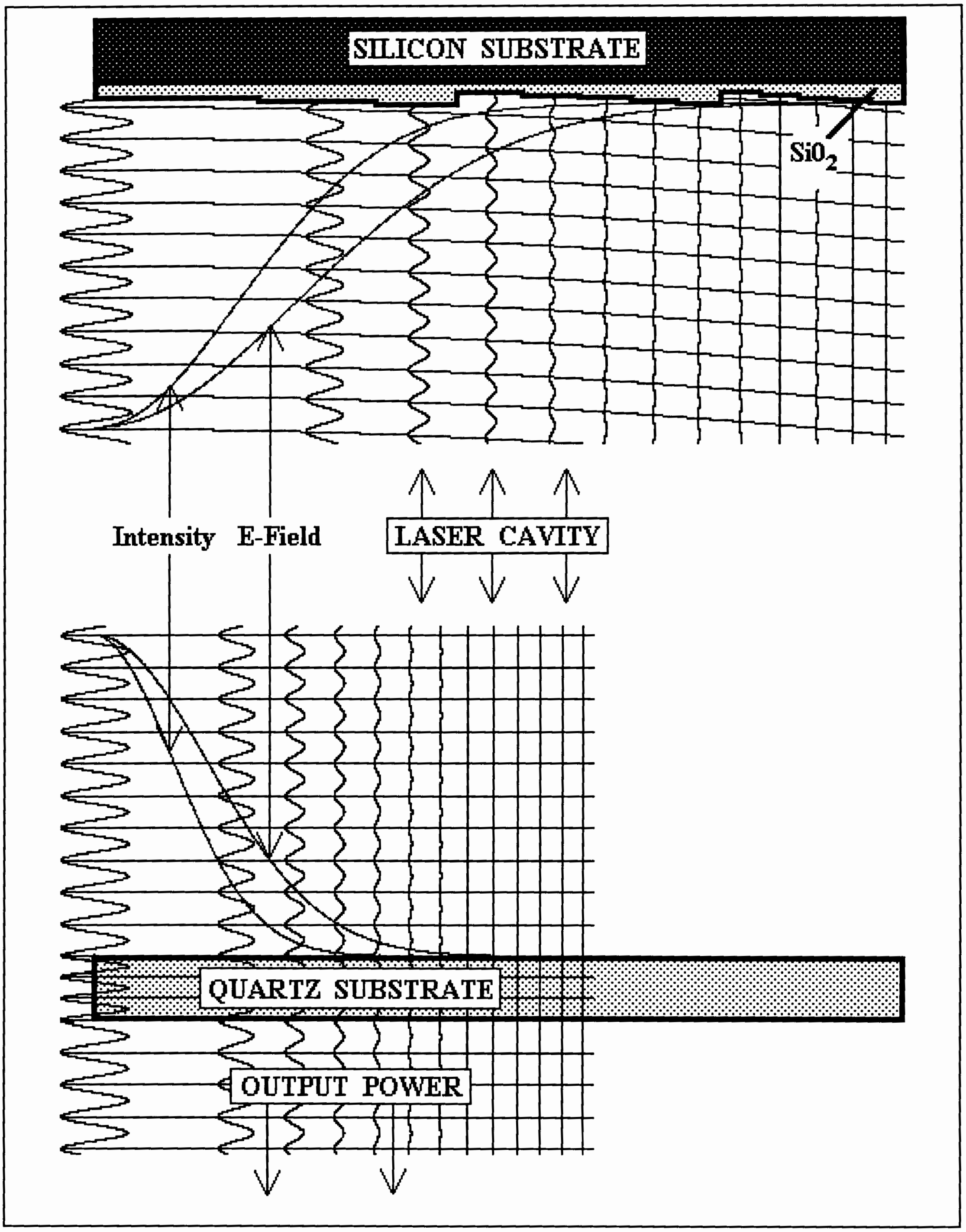


Figure 33. A radial cross-sectional view of a fully reflective Tiered(5) Fresnel Mirror with a partially transmitting Plane Mirror. Note: This is a Half-Symmetric Resonator.

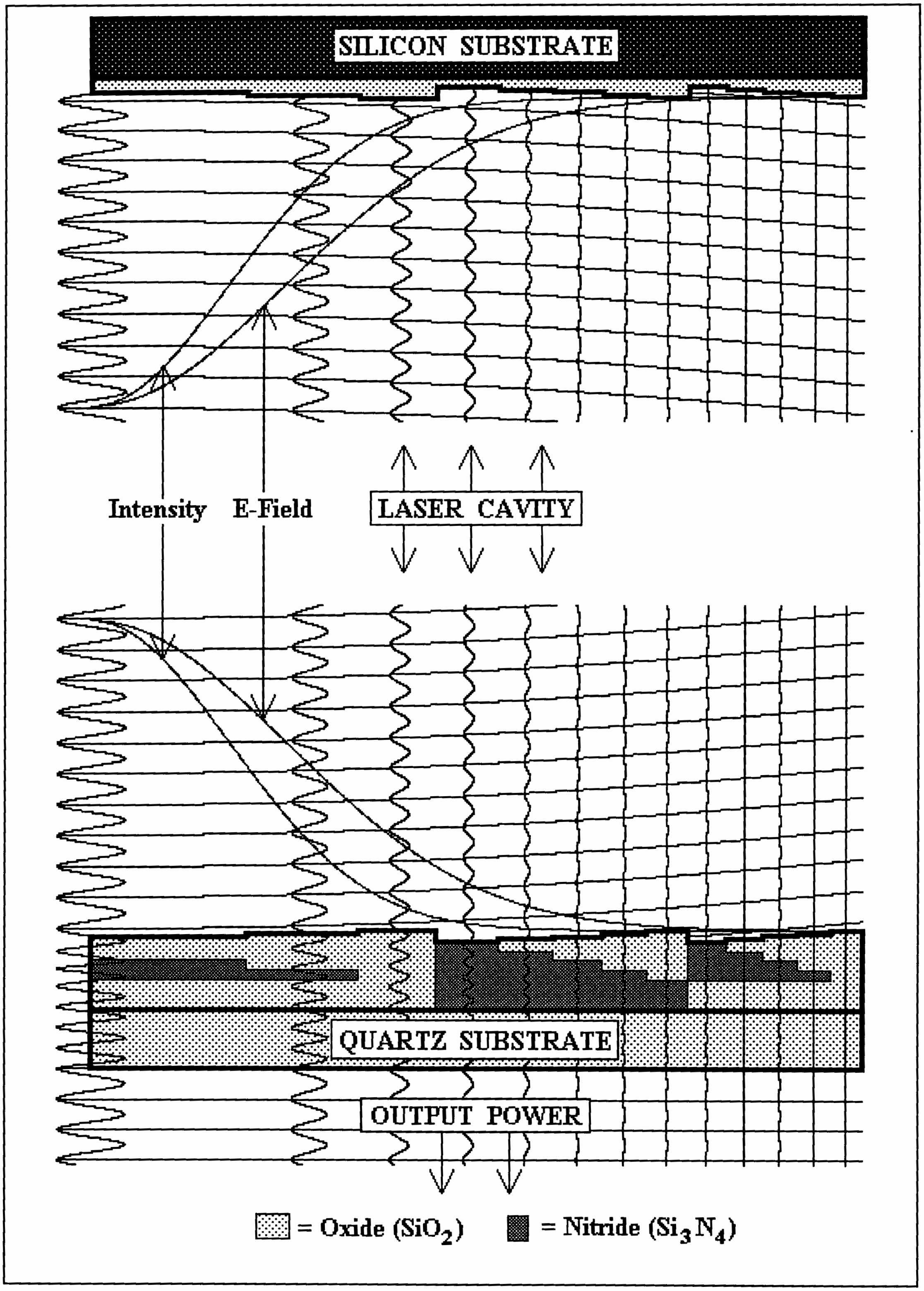


Figure 34. A radial cross-sectional view of a fully reflective and a partially transmitting Tiered(5) Fresnel Mirrors.

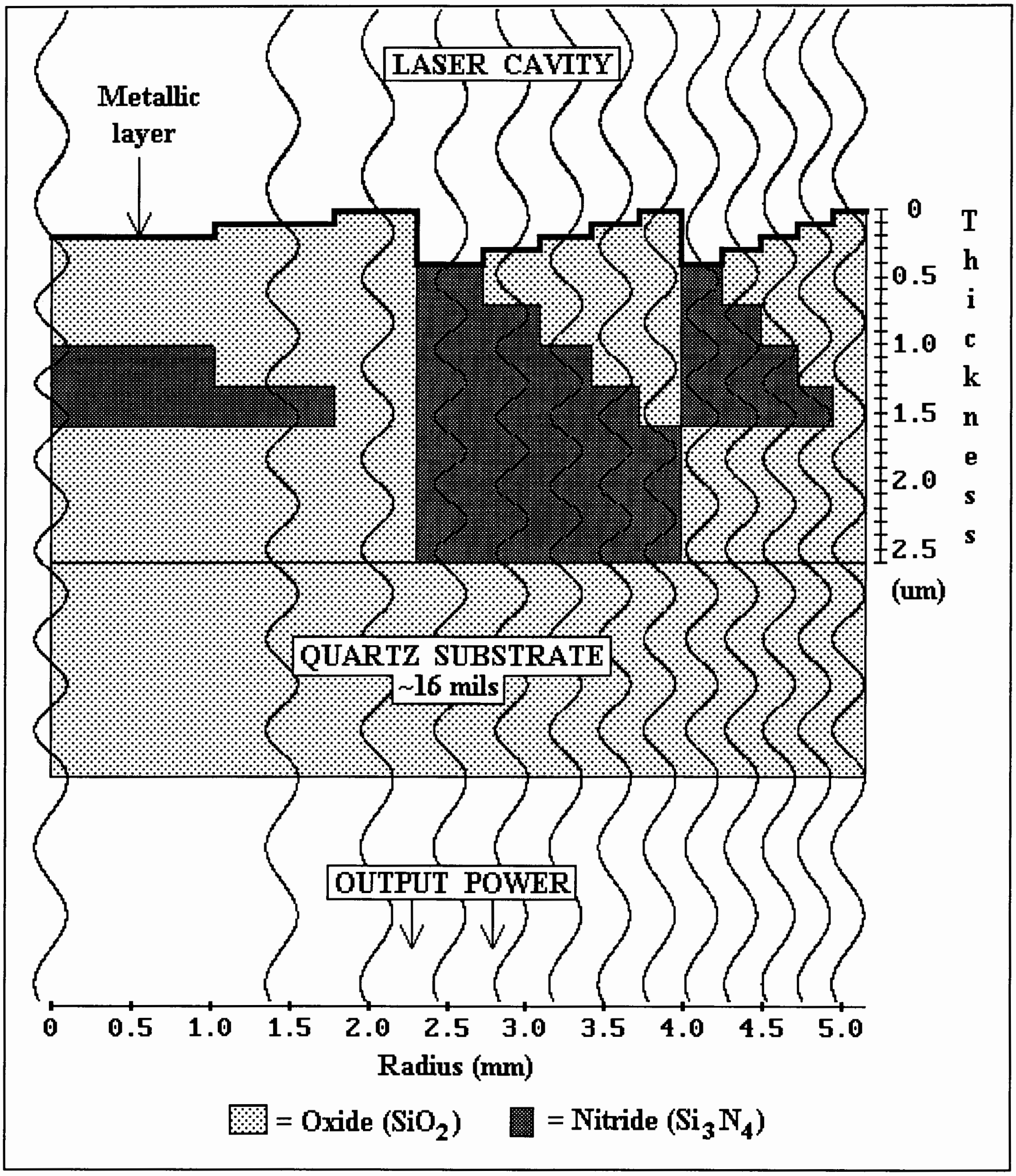


Figure 35. A zoom-in of the bottom mirror in Figure 34. The particular arrangement and thicknesses of the Oxide and Nitride film stacks convert a curved wavefront into a planar wavefront.

Figure 37 shows a sequence of cross-sections for the formation of the tiers from start to finish of a "five tiers per zone " reflective type Tiered Fresnel Mirror, i.e. a reflective type Tiered(5) Fresnel Mirror. A photomask is used to define the photoresist

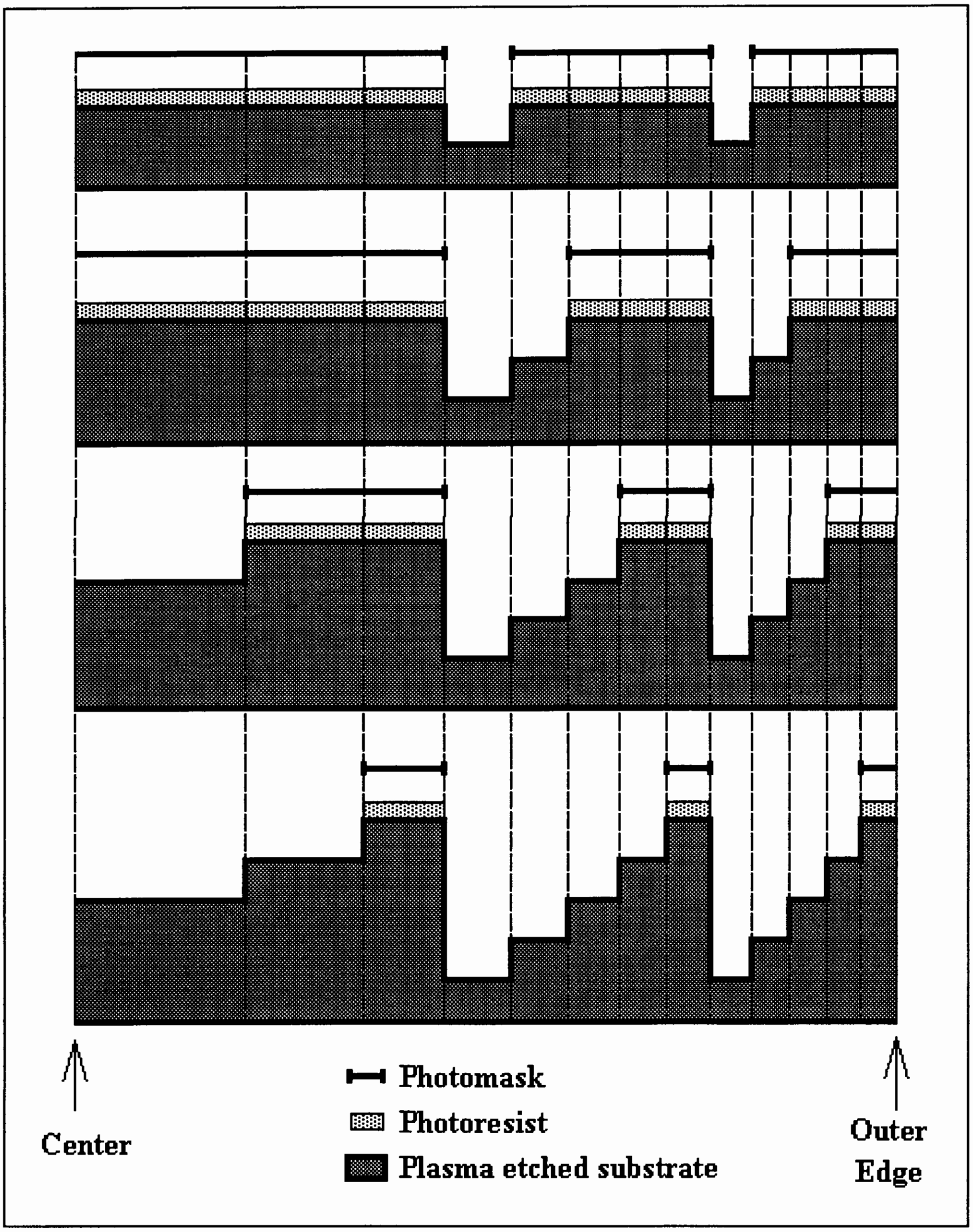


Figure 36. A process sequence, from start to finish (top to bottom), of radial cross-sections of the tier formation. A reflective type Tiered(5) Fresnel Mirror is shown.

pattern which is used to mask the etching of the substrate or some other layer. A Pattern-Etch cycle is required to define each tier. Therefore the cost of manufacturing is proportional to the number of tiers per zone of the Tiered Fresnel Mirror. The more tiers per zone the better the mirror performance yet the higher the cost to manufacture. The cost per performance must be determined prior to production since this defines the design of the Tiered Fresnel Mirror.

However, the main advantage of IC processing is the low cost per mirror realized due to batch processing where thousands of Tiered Fresnel Mirrors will be produced economically. Figure 37 shows a top view of an eight inch wafer having 185 Tiered(15) Fresnel Mirrors. Figure 38 shows a zoom-in of Figure 36 illustrating the individual tiers.

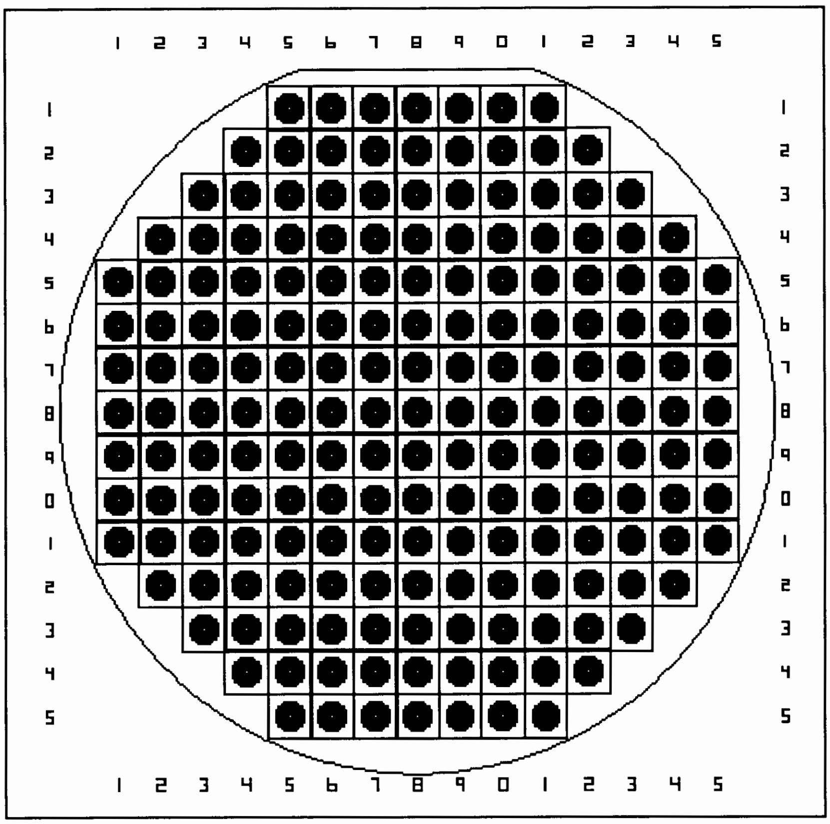


Figure 37. An eight inch wafer containing 185 Tiered(15) Fresnel Mirrors.

If a Lot of wafers consists of 20 eight inch wafers and if the yield is 90% one can obtain 3,330 Tiered(15) Fresnel Mirrors. If 10 Lots were processed one can obtain 33,300 Tiered(15) Fresnel Mirrors. So one can see that the cost per mirror decreases drastically from one wafer to a Lot of wafers. The price per Lot remains fixed. In other words, the cost to process one wafer is a little less than to process a whole Lot and the

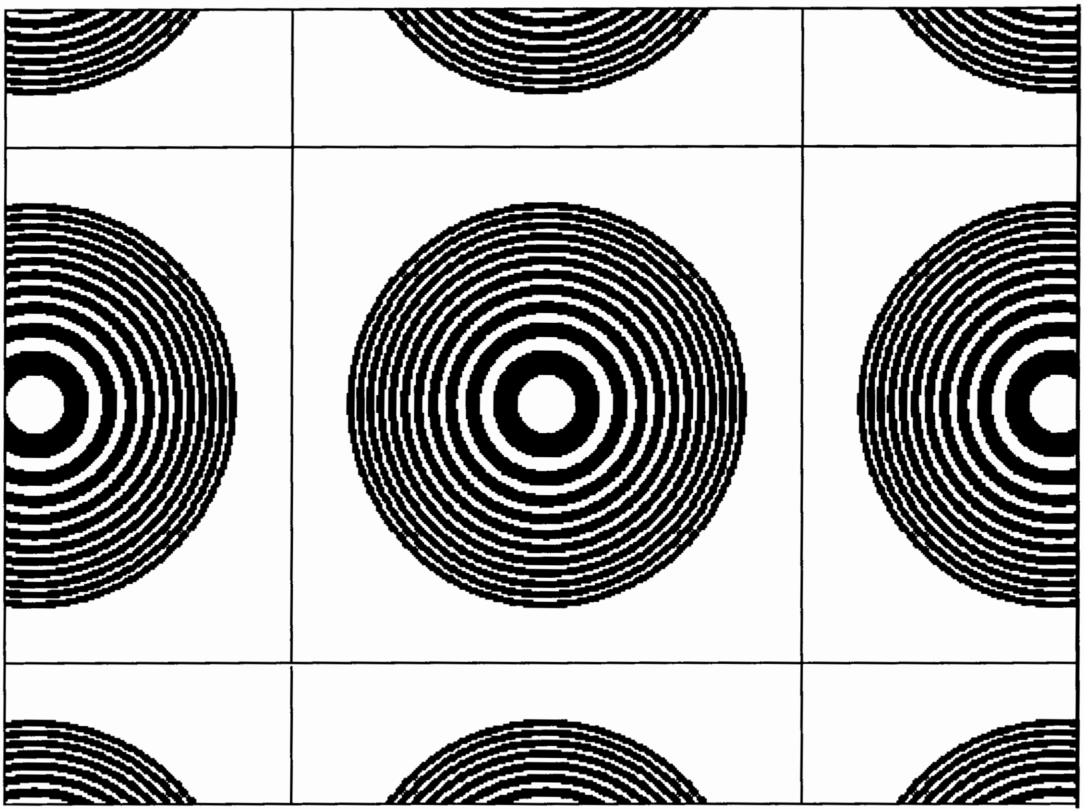


Figure 38. A zoom-in of a section of Figure 37. Notice there are 23 tiers in this three-zone Tiered(15) Fresnel Mirror.

one wafer yields say 166 mirrors whereas the Lot yields say 3,330 mirrors. The cost per mirror may be 10-20 times cheaper for the Lot compared to the single wafer.

A process simulation was performed where the variable of concern was the tier step height. A two zone Tiered(15) Fresnel Mirror was used to determine the performance in the form of a loss per pass value given in percent. A wavelength of 10.6 microns was used which determines the target tier step height at 3533 angstroms. An etch depth specification of 3533 ± 350 angstroms (3 σ) was incorporated into the simulation. Three separate trial runs were compared to the ideal case where all tier step heights were equal to 3533 angstroms. Table III shows the data for this simulation. Note that the loss per pass does not vary much which says that a 350 angstrom tolerance is satisfactory for the performance of the mirror. A ± 350 angstrom tolerance range is a very doable specification range for a plasma etch process step.

TABLE III

SIMULATED TIER HEIGHTS
WITH ITS AFFECT ON LOSS PER PASS
FOR A SYMMETRIC CONFOCAL RESONATOR

	Thicknesses in Angstroms						
	IDEAL	TRIAL 1		TRIAL 2		TRIAL 3	
TIER No.	**L & R** Mirrors	**Left** Mirror	**Right** Mirror	**Left** Mirror	**Right** Mirror	**Left** Mirror	**Right** Mirror
1	3533	3288	3463	3743	3288	3568	3463
2	3533	3463	3778	3498	3463	3638	3463
3	3533	3568	3848	3428	3253	3218	3253
4	3533	3673	3708	3498	3568	3288	3708
5	3533	3568	3288	3568	3638	3778	3463
6	3533	3218	3708	3778	3463	3673	3358
7	3533	3498	3638	3183	3358	3673	3603
8	3533	3498	3848	3288	3183	3498	3498
9	3533	3568	3568	3358	3673	3743	3603
10	3533	3253	3288	3393	3288	3708	3743
11	3533	3638	3708	3428	3253	3708	3603
12	3533	3463	3498	3288	3428	3498	3393
13	3533	3848	3673	3323	3218	3358	3568
14	3533	3708	3498	3568	3498	3463	3463
15	3533	3568	3603	3323	3288	3463	3393
16	3533	3288	3463	3743	3288	3568	3463
17	3533	3463	3778	3498	3463	3638	3463
18	3533	3568	3848	3428	3253	3218	3253
19	3533	3673	3708	3498	3568	3288	3708
20	3533	3568	3288	3568	3638	3778	3463
21	3533	3218	3708	3778	3463	3673	3358
22	3533	3498	3638	3183	3358	3673	3603
23	3533	3498	3848	3288	3183	3498	3498
LPP	1.535%	1.54%		1.56%		1.59%	

CHAPTER VI

CONCLUSION

The use of Tiered Fresnel Mirrors in Optical Resonators has been shown to be feasible. The appealling aspect of the low cost per mirror makes it desireable to produce. The Integrated Circuits Process has proven to be cost effective with Solid State Chips and for the same reason the Tiered Fresnel Mirror can be produced at a low cost per mirror. The performance was shown to be less than the Spherical Mirror yet the cost per performance ratio can be lower. In many instances one may settle for using a less efficient laser with the Tiered Fresnel Mirror depending upon the usage.

It was shown that the Fresnel Mirror acts like the Spherical Mirror and that the Tiered Fresnel Mirror is a modified Fresnel Mirror. The Tiered Fresnel Mirror can be made to act as a either a Spherical Mirror, a Plane Mirror, or somewhere between the two. How this is done is in the design, i.e. the number of tiers per zone. To emulate a Spherical Mirror one would use an infinite number of tiers per zone while on the other hand using zero tiers per zone emulates the Plane Mirror.

The Tiered Fresnel Mirror will naturally discriminate against higher transverse modes from oscillating in favor of the fundamental mode to a higher degree than that of the Spherical Mirror. With this in mind, it is easier to produce and maintain the fundamental mode and thus the TFM is an inherent mode discriminator.. Laser operation in the fundamental mode is desirable because it is generally more useful due to the beam's compact size and shape.

Finally the TFM is an inherent frequency or wavelength filter. The major variable in the design of the mirror is the wavelength from which the tier widths are determined.

Any deviance from this wavelength causes the performance to suffer. Thus the half-width of the Lorentzian lineshape function may be narrowed compared to that of a conventional spherical mirror system where less discrimination of the wavelength occurs.

REFERENCES

[1] Sobel, F., Wentworth, L. & Wiltse C., "Quasi-optical surface waveguide and other components for the 100 to 300-GHz region",
IRE Trans. on Microwave Theory Tech., Vol MTT-9, Nov 1961, pp 512-518.

[2] Black D. & Wiltse C., "Millimeter-Wave Characteristics of Phase-Correcting Fresnel Zone Plates",
IEEE Trans.on Microwave Theory and Techniques, Vol MTT-35, Dec 1987, pp 1122-1128.

[3] Haruna M., Takahaski M., Wakahayashi K. & Nioshihara H., "Laser Beam Lithographed Micro-Fresnel Lenses",
Applied Optics, Vol 29 No. 34, 1 Dec 1990, pp 5120-5126.

[4] Nishihara H., & Suhara T., "Micro Fresnel Lenses",
Progress in Optics XXIV, E. Wolf (Elsevier Science Publishers B.V., 1987).

[5] Mottier P. & Valette S., "Integrated Fresnel Lens on Thermally Oxidized Silicon Substrate",
Applied Optics, Vol 20, No 9, 1 May 1981, pp 1630-1634.

[6] Garrett J. & Wiltse J., "Fresnel Zone Plate Antennas at Millimeter Wavelengths",
International Journal of Infrared and Millimeter Waves, Vol 12, No 3, 1991, pp 195-221.

[7] Belanger P. & Pare C., "Optical Resonators Using Graded-Phase Mirrors",
Optics Letters, Vol 16, No 14, July 15, 1991, pp 1057-1059.

[8] Born M., & Wolf E., "Principles of Optics",
(Pergamon Press: New York, NY, 6th Ed., 1980) pg 262.

[9] Siegman, A.E., "Lasers",
(University Science Books: Mill Valley, Ca, 1986) pp581-593.

[10] Yariv, A., "Quantum Electronics",
(John Wiley & Sons: New York, N.Y., 3rd Ed., 1989) pp 106-109.

[11] Saleh, B.E.A., & Teich, M.C., "Fundamentals of Photonics",
(John Wiley & Sons: New York, N.Y., 1991) pp27-30.

[12] Yariv, A., pp 142-144.

[13] Siegman, A.E., pp642-646, 686.

[14] Saleh, B.E.A., & Teich, M.C., pp335-337.

[15] Siegman, A.E., pp647.

[16] Hecht,E. & Zajac, A., "Optics",
(Addison-Wesley Publishing Company, Inc.: Menlo Park, Ca.,1974) pp366-372.

[17] Sussman, M., "Elementary Diffraction Theory of Zone Plates",
American Journal of Physics, No. 28, 1960, pp394-398.

[18] Siegman, A.E., pg 690.

[19] Fox, A.G., & Li, T., "Resonant Modes in a Maser Interferometer",
The Bell System Technical Journal, 1961, pp453-488.

[20] Hildebrand, F.B., "Methods of Applied Mathematics",
(Prentice-Hall: Englewood Cliffs, N.J., 1952), pp 421-428.

APPENDIX A

LISTING OF COMPUTER PROGRAM "RESONATE VERSION 1.0"

```
!******************************************************** !
!                                                         !
!              PROGRAM:  RESONATE VERSION 1.0             !
!                                                         !
!        File Name: "RESONATE.U"          28 SEP 93       !
!                                                         !
!******************************************************** !
|
DIM z_1(1),z_2(1),zr1r2_1(1,1),zr1r2_2(1,1),r1r22_1(1,1),r1r22_2(1,1)
DIM r1_lambda_1(1),r2_lambda_2(1)
DIM gre(25),gim(25),gre2(51),gim2(51)
DIM loss1_per_pass(200),loss2_per_pass(200),loss_per_pass(400)
DIM intvar1_per_pass(100),intvar2_per_pass(100)
DIM re1(51),im1(51),re2(51),im2(51),step_rings(1)
DIM zz1(51),zz2(51),zzm(51),cos_t(25),phase_delay(1),depth1(1),depth2(1)
DIM rings1(10),rings2(10),rad1(1),rad2(1)
!DIM kr_n(1,1,51,25),h_n(1,1,51,25)
OPTION NOLET
|
!******************************************************** !
!         FUNCTION INPUT_NUM: Returns a number.           !
!******************************************************** !
|
DEF input_num(row,col,default)
| SET CURSOR row,col
| GET KEY key
| IF key=592 OR key=13 then                        ! dn arrow key or Cr.
| | IF rrow=9 OR rrow=16 then rrow=rrow+2 ELSE rrow=rrow+1
| | input_num=default
| | SET CURSOR row,col-1
| | PRINT default
| | EXIT DEF
| ELSE IF key=584 then                             ! up arrow key.
| | IF rrow=18 then rrow=rrow-2 ELSE rrow=rrow-1
| | input_num=default
| | SET CURSOR row,col-1
| | PRINT default
| | EXIT DEF
| ELSE IF key=583 then                             ! home key.
| | rrow=4
| | input_num=default
| | SET CURSOR row,col-1
| | PRINT default
| | EXIT DEF
```

```
| ELSE IF key=591 then                                   ! end key.
| | rrow=24
| | input_num=default
| | SET CURSOR row,col-1
| | PRINT default
| | EXIT DEF
| END IF
| WHEN ERROR IN
| | PRINT CHR$(key);
| | CALL input_
| | a$=CHR$(key) & a$
| | input_num=val(a$)
| | IF rrow=9 OR rrow=16 then rrow=rrow+2 ELSE rrow=rrow+1
| USE
| | IF EXTYPE=4001 then
| | | SET COLOR "black/white"
| | | SET CURSOR 2,1
| | | PRINT "Invalid # format, Cr to continue:";
| | | SET COLOR "white/black"
| | | CALL input_
| | | SET CURSOR 2,1
| | | PRINT erase_line$
| | | input_num=default
| | ELSE
| | | SET COLOR "black/white"
| | | SET CURSOR 2,1
| | | PRINT EXTYPE;EXTEXT$;", Cr to continue:";
| | | SET COLOR "white/black"
| | | CALL input_
| | | SET CURSOR 2,1
| | | PRINT erase_line$
| | | input_num=default
| | END IF
| END WHEN
END DEF
|
!************************************************************ !
!             FUNCTION INPUT_STRING: Returns a string.          !
!************************************************************ !
|
DEF input_string$(row,col,a1$,a2$,a3$,a4$,a5$,response$,default$)
| SET CURSOR row,col
| GET KEY key
| IF key=592 OR key=13 then                              ! dn arrow key or Cr.
```

```
| | rrow=rrow+1
| | input_string$=default$
| | PRINT default$
| | EXIT DEF
| ELSE IF key=584 then                                  ! up arrow key.
| | IF rrow=11 then rrow=rrow-2 ELSE rrow=rrow-1
| | input_string$=default$
| | PRINT default$
| | EXIT DEF
| ELSE IF key=583 then                                  ! home key.
| | rrow=4
| | input_string$=default$
| | PRINT default$
| | EXIT DEF
| ELSE IF key=591 then                                  ! end key.
| | rrow=24
| | input_string$=default$
| | PRINT default$
| | EXIT DEF
| END IF
| WHEN ERROR IN
| | PRINT CHR$(key);
| | CALL input_
| | a$=CHR$(key) & a$
| | a$=UCASE$(a$)
| | IF a$=a1$ OR a$=a2$ OR a$=a3$ OR a$=a4$ OR a$=a5$ then
| | | input_string$=a$
| | | rrow=rrow+1
| | ELSE
| | | SET COLOR "black/white"
| | | SET CURSOR 2,1
| | | PRINT response$;", Cr to continue:";
| | | SET COLOR "white/black"
| | | CALL input_
| | | SET CURSOR 2,1
| | | PRINT erase_line$
| | | input_string$=default$
| | END IF
| USE
| | SET COLOR "black/white"
| | SET CURSOR 2,1
| | PRINT EXTYPE;EXTEXT$;", Cr to continue:";
| | SET COLOR "white/black"
| | CALL input_
```

```
| | SET CURSOR 2,1
| | PRINT erase_line$
| | input_string$=default$
| END WHEN
END DEF
|
!******************************************************* !
!    SUBROUTINE INPUTS: Retains current inputs after CNTRL-BRK.  !
!******************************************************* !
|
SUB inputs                    ! If one CNTRL_BRK's the program while running,
| inputs$="Y"                 ! at the command line, type "CALL INPUTS" to
| CALL main                   ! re-run the program with the current inputs.
END SUB
|
!******************************************************* !
!       SUBROUTINE OPEN_VIEWPORTS: Opens screen viewports.    !
!******************************************************* !
|
SUB open_viewports
|
| FOR i=1 to 15
| | CLOSE # i
| NEXT i
| OPEN #1 : SCREEN 0.0500,0.9500,0,1          !!!
| OPEN #2 : SCREEN 0.0500,0.1100,0.45,0.8       !
| OPEN #3 : SCREEN 0.1100,0.2753,0.45,0.8       !
| OPEN #4 : SCREEN 0.3113,0.4766,0.45,0.8       !
| OPEN #5 : SCREEN 0.4994,0.6647,0.45,0.8       !
| OPEN #6 : SCREEN 0.7247,0.8900,0.45,0.8       !
| OPEN #7 : SCREEN 0.890,0.950,0.45,0.8         ! Screen
| OPEN #8 : SCREEN 0.110,0.338,0.08,0.4         ! viewports.
| OPEN #9 : SCREEN 0.374,0.602,0.08,0.4         !
| OPEN #10: SCREEN 0.662,0.890,0.08,0.4         !
| OPEN #11: SCREEN 0.110,0.338,0.04,0.07        !
| OPEN #12: SCREEN 0.374,0.602,0.04,0.07        !
| OPEN #13: SCREEN 0.662,0.890,0.04,0.07        !
| OPEN #14: SCREEN 0.090,0.935,0.815,0.835      !
| OPEN #15: SCREEN 0.05,0.95,0,0.04           !!!
END SUB
|
!******************************************************* !
!          SUBROUTINE INIT: Initializes the important variables.          !
!******************************************************* !
```

```
|
SUB init
| CALL open_viewports
| ! Mirror types: SPHERICAL, PARABOLIC, PLANE, FRESNEL, or TIERED.
| m1$="SPHERICAL"                        ! Mirror1 type.
| m2$="SPHERICAL"                        ! Mirror2 type.
| radius1,radius2=1                      ! Radii of curvature for M1,M2.
| zones1,zones2=3                        ! # of complete Fresnel zones.
| tpz1,tpz2=15                   ! # of tiers per zone. Must be ODD.
| outer_tiers1=0                 ! # of outer-zone tiers, counting from the
| outer_tiers2=0                 !                center of the last zone.
| cavity_length=1                ! The resonator length (mirror separation).
| lambda=10.6e-6                 ! The wavelength of the laser light.
| max_rt=200                     ! Total transits (2 transits=1 round trip).
| incr=50         ! Radial increments. MUST BE EVEN for SUB Simpsons_Rule.
| inct=40         ! Theta  increments. MUST BE EVEN for SUB Simpsons_Rule.
| input_wave$="PLANE"            ! "PLANE" (or "GAUSSIAN"): Input wave.
| IF MOD(tpz1,2)=0 then tpz1=tpz1+1          ! Odd # of tiers per zone only.
| IF zones1=0 then
| | tiers1=outer_tiers1
| ELSE
| | tiers1=tpz1*(zones1-0.5)+0.5+outer_tiers1              ! Total # of tiers.
| END IF
| max_radius1=SQR(radius1*(2*tiers1-2)*lambda/2/tpz1+((2*tiers1-2)
| | *lambda/2/tpz1)^2)
| IF MOD(tpz2,2)=0 then tpz2=tpz2+1          ! Odd # of tiers per zone only.
| IF zones2=0 then
| | tiers2=outer_tiers2
| ELSE
| | tiers2=tpz2*(zones2-0.5)+0.5+outer_tiers2              ! Total # of tiers.
| END IF
| max_radius2=SQR(radius2*(2*tiers2-2)*lambda/2/tpz2+((2*tiers2-2)
| | *lambda/2/tpz2)^2)
| similate$="N"           ! Y/N: Similates a processing of the step heights.
| RANDOMIZE               ! Used for process similation.
| alternate$="Y"          ! Step mirrors: M1=(recessed,raised) M2=vice versa.
| step_switch$="N"        ! Step mirrors: Recessed->raised;raised->recessed.
| step_tiers=5            ! Step mirrors only. Total # of rings.
| n=0                     ! Used to maximize RAM usage.
| show$="Y"
| dev=7e-4         ! Stops program when the amplitude fluxuations are minimal.
| auto_scale$="Y"                ! Y/N: Auto-scales the log plot of power loss.
| erase_line$=REPEAT$(" ",77)
END SUB
```

```
|
!**********************************************************!
!              PROGRAM STARTS HERE: Begin the program.              !
!**********************************************************!
|
inputs$="N"
CALL main
|
!**********************************************************!
!                SUBROUTINE MAIN: The main subroutine.               !
!**********************************************************!
|
SUB main
| IF inputs$="N" then CALL init
| DO
| | CALL open_viewports
| | WINDOW #1
| | CLEAR
| | SET COLOR "white"
| | CALL variable_change
| | IF MOD(tpz1,2)=0 then tpz1=tpz1+1
| | tiers1=tpz1*zones1                          ! Number of tiers.
| | IF zones1=0 then
| | | tiers1=outer_tiers1
| | ELSE                                        ! Total tiers=tiers+
| | | tiers1=tiers1-(tpz1-1)/2+outer_tiers1     !      outer_tiers.
| | END IF
| | IF MOD(tpz2,2)=0 then tpz2=tpz2+1
| | tiers2=tpz2*zones2                          ! Number of tiers.
| | IF zones2=0 then
| | | tiers2=outer_tiers2
| | ELSE                                        ! Total tiers=tiers+
| | | tiers2=tiers2-(tpz2-1)/2+outer_tiers2     !      outer_tiers.
| | END IF
| | max_transits=2*max_rt                       ! Total transits or passes.
| | MAT REDIM z_1(incr+1),z_2(incr+1)
| | MAT REDIM zr1r2_1(incr+1,incr+1),zr1r2_2(incr+1,incr+1)
| | MAT REDIM r1_lambda_1(incr+1),r2_lambda_2(incr+1)
| | MAT REDIM r1r22_1(incr+1,incr+1),r1r22_2(incr+1,incr+1)
| | MAT REDIM rad1(zones1+1),rad2(zones2+1)
| | MAT REDIM rings1(tiers1),rings2(tiers2)
| | MAT REDIM phase_delay(incr+1),depth1(tpz1),depth2(tpz2)
| | MAT REDIM gre(inct+1),gim(inct+1),gre2(incr+1),gim2(incr+1)
| | MAT REDIM re1(incr+1),im1(incr+1),re2(incr+1),im2(incr+1)
```

```
| | MAT REDIM zz1(incr+1),zz2(incr+1),zzm(incr+1),cos_t(inct+1)
| | MAT REDIM loss1_per_pass(max_rt),loss2_per_pass(max_rt)
| | MAT REDIM step_rings(step_tiers+1),loss_per_pass(max_transits)
| | MAT REDIM intvar1_per_pass(max_rt),intvar2_per_pass(max_rt)
| | !MAT REDIM kr_n(2,n+1,incr+1,inct+1),h_n(2,n+1,incr+1,inct+1)
| | input_wave$=UCASE$(input_wave$)
| | m1$=UCASE$(m1$)
| | m2$=UCASE$(m2$)
| | alternate$=UCASE$(alternate$)
| | step_switch$=UCASE$(step_switch$)
| | auto_scale$=UCASE$(auto_scale$)
| | show=0.845/incr
| | k=2*pi/lambda
| | IF m1$="STEP" then
| | | etch_depth=lambda/4
| | | etch_depth$="LAMBDA/4"
| | | FOR i=1 to step_tiers                              ! Find the odd lambda/4
| | | | j=2*i-1                                          ! phase incremented radii.
| | | | | step_rings(i)=SQR(fzt_focus*j*lambda/2+(j*lambda/4)^2)
| | | NEXT i
| | | step_max_radius=step_rings(step_tiers)                 ! Max mirror radius.
| | | dr1,dr2=step_max_radius/incr                           ! Incremental radii.
| | ELSEIF m1$="TIERED" OR m1$="FRESNEL" then
| | | FOR i=1 to tiers1                           ! Find odd lambda/(2*tpz1) radii.
| | | | IF i=tiers1 then
| | | | | | j=2*i-2                          ! The outside edge of the outermost tier.
| | | | ELSE
| | | | | | j=2*i-1
| | | | END IF
| | | | | |  ! j=2*i-1                             ! The center of the outermost tier.
| | | | rings1(i)=SQR(radius1*j*lambda/2/tpz1+(j*lambda/2/tpz1)^2)
| | | NEXT i
| | | max_radius1=rings1(tiers1)                            ! Max Mirror1 radius.
| | | FOR i=1 to zones1                  ! For F,T mirrors: Radii of Fresnel zones.
| | | | rad1(i)=SQR((2*i-1)*lambda*radius1/2+((2*i-1)*lambda/4)^2)
| | | NEXT i
| | | rad1(i)=max_radius1
| | | etch_depth1=lambda/(2*tpz1)
| | | etch_depth1$="W/" & STR$(2*tpz1)
| | | tol=700e-10                  ! Tolerance of etch_depth1 = +/- 700 Angstroms.
| | | sim1=tol/10                  ! Used to divide the tolerance range into 20 parts.
| | | sim$=""                   ! A=-tol, B=-0.9tol..., J=-0.1tol, K=+0.1tol..., T=+tol.
| | | scale=1.5                 ! Used to convert a uniform into a normal distribution.
| | | FOR i=1 to tpz1
```

```
| | | | IF similate$="N" then
| | | | | depth1(i)=etch_depth1*MOD((tpz1-3)/2+i,tpz1)
| | | | ELSE
| | | | | random=10*rnd
| | | | | IF random < 5 then
| | | | | | sim=tol/scale*(scale-LOG(1+random*(EXP(scale)-1)/5))
| | | | | | sim=-sim1*INT(1+sim/sim1)                    ! - deviance from
| | | | | | s$=CHR$(75+INT(sim/sim1))                    ! mean (thinner).
| | | | | ELSE
| | | | | | random=10-random
| | | | | | sim=tol/scale*(scale-LOG(1+random*(EXP(scale)-1)/5))
| | | | | | sim=sim1*INT(1+sim/sim1)                     ! + deviance from
| | | | | | s$=CHR$(74+INT(sim/sim1))                    ! mean (thicker).
| | | | | END IF
| | | | | sim$=sim$ & s$
| | | | | depth1(i)=etch_depth1*MOD((tpz1-3)/2+i,tpz1)+sim
| | | | END IF
| | | NEXT i
| | END IF
| | dr1=max_radius1/incr                    ! Incremental radius: dr1 for Mirror1.
| | |
| | IF m2$="STEP" then
| | | etch_depth=lambda/4
| | | etch_depth$="LAMBDA/4"
| | | FOR i=1 to step_tiers                          ! Find the odd lambda/4
| | | | j=2*i-1                                      ! phase incremented radii.
| | | | step_rings(i)=SQR(fzt_focus*j*lambda/2+(j*lambda/4)^2)
| | | NEXT i
| | | step_max_radius=step_rings(step_tiers)         ! Max mirror radius.
| | | dr1,dr2=step_max_radius/incr                   ! Incremental radii.
| | ELSEIF m2$="TIERED" OR m2$="FRESNEL" then
| | | FOR i=1 to tiers2                     ! Find odd lambda/(2*tpz2) radii.
| | | | IF i=tiers2 then
| | | | | j=2*i-2                           ! The outer edge of the outermost tier.
| | | | ELSE
| | | | | j=2*i-1
| | | | END IF
| | | | |  ! j=2*i-1                        ! The center of the outermost tier.
| | | | rings2(i)=SQR(radius2*j*lambda/2/tpz2+(j*lambda/2/tpz2)^2)
| | | NEXT i
| | | max_radius2=rings2(tiers2)                          ! Max Mirror1 radius.
| | | FOR i=1 to zones2               ! For F,T mirrors: Radii of Fresnel zones.
| | | | rad2(i)=SQR((2*i-1)*lambda*radius2/2+((2*i-1)*lambda/4)^2)
| | | NEXT i
```

```
| | | rad2(i)=max_radius2
| | | etch_depth2=lambda/(2*tpz2)
| | | etch_depth2$="W/" & STR$(2*tpz2)
| | | tol=700e-10                ! Tolerance of etch_depth2 = +/- 700 Angstroms.
| | | sim1=tol/10                ! Used to divide the tolerance range into 20 parts.
| | | sim$=""                ! A=-tol, B=-0.9tol..., J=-0.1tol, K=+0.1tol..., T=+tol.
| | | scale=1.5              ! Used to convert a uniform into a semi-normal
| | | FOR i=1 to tpz2            !                             distribution.
| | | | IF similate$="N" then
| | | | | depth2(i)=etch_depth2*MOD((tpz2-3)/2+i,tpz2)
| | | | ELSE
| | | | | random=10*rnd
| | | | | IF random < 5 then
| | | | | | sim=tol/scale*(scale-LOG(1+random*(EXP(scale)-1)/5))
| | | | | | sim=-sim1*INT(1+sim/sim1)                    ! - deviance from
| | | | | | s$=CHR$(75+INT(sim/sim1))                    ! mean (thinner).
| | | | | ELSE
| | | | | | random=10-random
| | | | | | sim=tol/scale*(scale-LOG(1+random*(EXP(scale)-1)/5))
| | | | | | sim=sim1*INT(1+sim/sim1)                     ! + deviance from
| | | | | | s$=CHR$(74+INT(sim/sim1))                    ! mean (thicker).
| | | | | END IF
| | | | | sim$=sim$ & s$
| | | | | depth2(i)=etch_depth2*MOD((tpz2-3)/2+i,tpz2)+sim
| | | | END IF
| | | NEXT i
| | END IF
| | dr2=max_radius2/incr                  ! Incremental radius: dr2 for Mirror2.
| | |
| | dt=pi/inct                                          ! Incremental theta.
| | FOR i=0 to inct
| | | cos_t(i+1)=cos(i*dt)
| | NEXT i
| | |
| | WINDOW #2                                              ! Plot mirror1.
| | IF max_radius1 => max_radius2 then
| | | bot_top=incr
| | ELSE
| | | bot_top=max_radius2/max_radius1*incr
| | END IF
| | IF radius1 > 0 then
| | | left=-0.15
| | | right=0.05
| | | flood_cent=-0.125
```

```
| | ELSE
| | | left=-0.05
| | | right=0.15
| | | flood_cent=0
| | END IF
| | SET WINDOW left,right,-bot_top,bot_top
| | SET COLOR 10                                   ! "intensified green"
| | IF m1$="STEP" then
| | | FOR i=0 to incr                 ! zz1 = Distance from refer.plane to mirror1.
| | | | j=i*dr1
| | | | FOR a=1 to step_tiers STEP 2
| | | | | IF j<=step_rings(a) then
| | | | | | IF step_switch$="Y" then zz1(i+1)=0 ELSE zz1(i+1)=etch_depth
| | | | | | EXIT FOR
| | | | | ELSEIF j<=step_rings(a+1) then
| | | | | | IF step_switch$="Y" then zz1(i+1)=etch_depth ELSE zz1(i+1)=0
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000                       ! Plot zz1.
| | | | j=i*dr1
| | | | FOR a=1 to step_tiers STEP 2
| | | | | IF ABS(j)<=step_rings(a) then
| | | | | | IF step_switch$="Y" then
| | | | | | | PLOT -0.1,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.1,-i
| | | | | | | | PLOT -0.1,i;
| | | | | | | END IF
| | | | | | ELSE
| | | | | | | PLOT -0.05,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.05,-i
| | | | | | | | PLOT -0.05,i;
| | | | | | | END IF
| | | | | | END IF
| | | | | | EXIT FOR
| | | | | ELSEIF ABS(j)<=step_rings(a+1) then
| | | | | | IF step_switch$="Y" then
| | | | | | | PLOT -0.05,i;
| | | | | | | IF i=-incr then
```

```
| | | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | | PLOT -0.05,-i
| | | | | | | | | PLOT -0.05,i;
| | | | | | | | END IF
| | | | | | | ELSE
| | | | | | | | PLOT -0.1,i;
| | | | | | | | IF i=-incr then
| | | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | | PLOT -0.1,-i
| | | | | | | | | PLOT -0.1,i;
| | | | | | | | END IF
| | | | | | | END IF
| | | | | | | EXIT FOR
| | | | | | END IF
| | | | NEXT a
| | | NEXT i
| | ELSEIF m1$="PLANE" then
| | | MAT zz1=0
| | | PLOT -0.05,incr;-0.05,-incr;-0.15,-incr;-0.15,incr;-0.05,incr
| | ELSEIF m1$="TIERED" then
| | | FOR i=0 to incr
| | | | j=i*dr1
| | | | FOR a=1 to tiers1
| | | | | IF j<=rings1(a) then
| | | | | | CALL DIVIDE(a,tpz1,q,q1)
| | | | | | IF q1=0 then q1=tpz1
| | | | | | zz1(i+1)=depth1(q1)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr1
| | | | FOR a=1 to tiers1
| | | | | IF ABS(j)<=rings1(a) then                    ! q1=remainder of a/tpz1.
| | | | | | CALL DIVIDE(a,tpz1,q,q1)
| | | | | | IF q1=0 then q1=tpz1
| | | | | | PLOT -0.1*(1-depth1(q1)/depth1((tpz1+1)/2)),i;
| | | | | | IF i=-incr then
| | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | PLOT -0.1*(1-depth1(q1)/depth1((tpz1+1)/2)),-i
| | | | | | | PLOT -0.1*(1-depth1(q1)/depth1((tpz1+1)/2)),i;
| | | | | | END IF
| | | | | | EXIT FOR
```

```
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | ELSEIF m1$="SPHERICAL" OR m1$="PARABOLIC" then
| | | daf=(incr*dr1)^2
| | | p2=2*(radius1/2+SQR((radius1/2)^2+daf))
| | | FOR i=0 to incr
| | | | j=i*dr1
| | | | IF m1$="SPHERICAL" then
| | | | | zz1(i+1)=radius1 - SGN(radius1)*SQR((radius1)^2-j^2)
| | | | ELSE
| | | | | zz1(i+1)=(daf-j^2)/p2                              ! Parabolic.
| | | | END IF
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr1
| | | | IF m1$="SPHERICAL" then
| | | | | PLOT -0.1/zz1(incr+1)*((SQR((radius1)^2-j^2)-SQR((radius1)^2
| | | | | | -daf))),i;
| | | | | IF i=-incr then
| | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | PLOT -0.1/zz1(incr+1)*((SQR((radius1)^2-j^2)
| | | | | | | -SQR((radius1)^2-daf))),-i
| | | | | | PLOT -0.1/zz1(incr+1)*((SQR((radius1)^2-j^2)-SQR((radius1)^2
| | | | | | | -daf))),i;
| | | | | END IF
| | | | ELSE                                                 ! Parabolic mirror.
| | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),i;
| | | | | IF i=-incr then
| | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),-i
| | | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),i;
| | | | | END IF
| | | | END IF
| | | NEXT i
| | ELSEIF m1$="FRESNEL" then
| | | FOR i=0 to incr
| | | | j=i*dr1
| | | | FOR a=1 to zones1+1
| | | | | IF j<=rad1(a)+1e-7 then
| | | | | | zz1(i+1)=radius1-SQR(((a-1)*lambda/2+radius1)^2-j^2)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
```

```
| | | NEXT i
| | | z1=100
| | | z11=-100
| | | FOR i=1 to incr+1
| | | | z1=min(z1,zz1(i))
| | | | z11=max(z11,zz1(i))
| | | NEXT i
| | | MAT zzm=(-z1)*con(incr+1)
| | | MAT zz1=zzm+zz1
| | | z1=100
| | | z11=-100
| | | FOR i=0 to incr STEP incr/1000
| | | | j=i*dr1
| | | | FOR a=1 to zones1+1
| | | | | IF j<=rad1(a)+1e-7 then
| | | | | | ii=radius1-SQR(((a-1)*lambda/2+radius1)^2-j^2)
| | | | | | z1=min(z1,ii)
| | | | | | z11=max(z11,ii)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr1
| | | | FOR a=1 to zones1+1
| | | | | IF ABS(j)<=rad1(a)+1e-7 then
| | | | | | PLOT -0.1*(1-(radius1 - (SQR(((a-1)*lambda/2+radius1)^2-j^2)
| | | | | | | )-z1)/(z11-z1)),i;
| | | | | | IF i=-incr then
| | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | PLOT -0.1*(1-(radius1 - (SQR(((a-1)*lambda/2+radius1)^2
| | | | | | | | -j^2))-z1)/(z11-z1)),-i
| | | | | | | PLOT -0.1*(1-(radius1 - (SQR(((a-1)*lambda/2+radius1)^2
| | | | | | | | -j^2))-z1)/(z11-z1)),i;
| | | | | | END IF
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | END IF
| | FLOOD flood_cent,0
| | PLOT
| | |
| | WINDOW #7                                              ! Plot mirror2.
```

```
| | IF max_radius2 => max_radius1 then
| | | bot_top=incr
| | ELSE
| | | bot_top=max_radius1/max_radius2*incr
| | END IF
| | IF radius2 > 0 then
| | | left=0.05
| | | right=-0.15
| | | flood_cent=-0.125
| | ELSE
| | | left=0.15
| | | right=-0.05
| | | flood_cent=0
| | END IF
| | SET WINDOW left,right,-bot_top,bot_top
| | SET COLOR 11                                    ! "intensified cyan"
| | IF m2$="STEP" then
| | | FOR i=0 to incr              ! zz2 = Distance from refer. plane to mirror2.
| | | | j=i*dr2
| | | | FOR a=1 to step_tiers STEP 2
| | | | | IF j<=step_rings(a) then
| | | | | | IF (step_switch$="Y" and alternate$="Y") OR (step_switch$="N"
| | | | | | |  and alternate$="N") then zz2(i+1)=etch_depth ELSE zz2(i+1)=0
| | | | | | EXIT FOR
| | | | | ELSEIF j<=step_rings(a+1) then
| | | | | | IF (step_switch$="Y" and alternate$="Y") OR (step_switch$="N"
| | | | | | |  and alternate$="N") then zz2(i+1)=0 ELSE zz2(i+1)=etch_depth
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000                         ! Plot zz2.
| | | | j=i*dr2
| | | | FOR a=1 to step_tiers STEP 2
| | | | | IF ABS(j)<=step_rings(a) then
| | | | | | IF (step_switch$="Y" and alternate$="Y") OR (step_switch$="N"
| | | | | | |  and alternate$="N") then
| | | | | | | PLOT -0.05,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.05,-i
| | | | | | | | PLOT -0.05,i;
| | | | | | | END IF
| | | | | | ELSE
```

```
| | | | | | | PLOT -0.1,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.1,-i
| | | | | | | | PLOT -0.1,i;
| | | | | | | END IF
| | | | | | END IF
| | | | | | EXIT FOR
| | | | | ELSEIF ABS(j)<=step_rings(a+1) then
| | | | | | IF (step_switch$="Y" and alternate$="Y") OR (step_switch$="N"
| | | | | | |  and alternate$="N") then
| | | | | | | PLOT -0.1,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.1,-i
| | | | | | | | PLOT -0.1,i;
| | | | | | | END IF
| | | | | | ELSE
| | | | | | | PLOT -0.05,i;
| | | | | | | IF i=-incr then
| | | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | | PLOT -0.05,-i
| | | | | | | | PLOT -0.05,i;
| | | | | | | END IF
| | | | | | END IF
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | ELSEIF m2$="PLANE" then
| | | MAT zz2=0
| | | PLOT -0.05,incr;-0.05,-incr;-0.15,-incr;-0.15,incr;-0.05,incr
| | ELSEIF m2$="TIERED" then
| | | FOR i=0 to incr
| | | | j=i*dr2
| | | | FOR a=1 to tiers2
| | | | | IF j<=rings2(a) then                          ! q1=remainder of a/tpz2.
| | | | | | CALL DIVIDE(a,tpz2,q,q1)
| | | | | | IF q1=0 then q1=tpz2
| | | | | | zz2(i+1)=depth2(q1)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
```

```
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr2
| | | | FOR a=1 to tiers2
| | | | | IF ABS(j)<=rings2(a) then
| | | | | | CALL DIVIDE(a,tpz2,q,q1)
| | | | | | IF q1=0 then q1=tpz2
| | | | | | PLOT -0.1*(1-depth2(q1)/depth2((tpz2+1)/2)),i;
| | | | | | IF i=-incr then
| | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | PLOT -0.1*(1-depth2(q1)/depth2((tpz2+1)/2)),-i
| | | | | | | PLOT -0.1*(1-depth2(q1)/depth2((tpz2+1)/2)),i;
| | | | | | END IF
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | ELSEIF m2$="SPHERICAL" OR m2$="PARABOLIC" then
| | | daf=(incr*dr2)^2
| | | p2=2*(radius2/2+SQR((radius2/2)^2+daf))
| | | FOR i=0 to incr
| | | | j=i*dr2
| | | | IF m2$="SPHERICAL" then
| | | | | zz2(i+1)=radius2 - SGN(radius2)*SQR((radius2)^2-j^2)
| | | | ELSE                                                    ! Parabolic.
| | | | | zz2(i+1)=(daf-j^2)/p2
| | | | END IF
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr2
| | | | IF m2$="SPHERICAL" then
| | | | | PLOT -0.1/zz2(incr+1)*((SQR((radius2)^2-j^2)-SQR((radius2)^2
| | | | | | -daf))),i;
| | | | | IF i=-incr then
| | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | PLOT -0.1/zz2(incr+1)*((SQR((radius2)^2-j^2)-SQR((radius2)^2
| | | | | | | -daf))),-i
| | | | | | PLOT -0.1/zz2(incr+1)*((SQR((radius2)^2-j^2)-SQR((radius2)^2
| | | | | | | -daf))),i;
| | | | | END IF
| | | | ELSE                                                    ! Parabolic.
| | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),i;
| | | | | IF i=-incr then
| | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),-i
```

```
| | | | | | PLOT -0.1/zz1(1)*((daf-j^2)/p2),i;
| | | | | END IF
| | | | END IF
| | | NEXT i
| | ELSEIF m2$="FRESNEL" then
| | | FOR i=0 to incr
| | | | j=i*dr2
| | | | FOR a=1 to zones2+1
| | | | | IF j<=rad2(a)+1e-7 then
| | | | | | zz2(i+1)=radius2 - SQR(((a-1)*lambda/2+radius2)^2-j^2)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | z2=100
| | | z22=-100
| | | FOR i=1 to incr+1
| | | | z2=min(z2,zz2(i))
| | | | z22=max(z22,zz2(i))
| | | NEXT i
| | | MAT zzm=(-z2)*con(incr+1)
| | | MAT zz2=zzm+zz2
| | | z1=100
| | | z11=-100
| | | FOR i=0 to incr STEP incr/1000
| | | | j=i*dr2
| | | | FOR a=1 to zones2+1
| | | | | IF j<=rad2(a)+1e-7 then
| | | | | | ii=radius2 - SQR(((a-1)*lambda/2+radius2)^2-j^2)
| | | | | | z1=min(z1,ii)
| | | | | | z11=max(z11,ii)
| | | | | | EXIT FOR
| | | | | END IF
| | | | NEXT a
| | | NEXT i
| | | FOR i=-incr to incr STEP incr/1000
| | | | j=i*dr2
| | | | FOR a=1 to zones2+1
| | | | | IF ABS(j)<=rad2(a)+1e-7 then
| | | | | | PLOT -0.1*(1-(radius2 - (SQR(((a-1)*lambda/2+radius2)^2-j^2))
| | | | | | | -z1)/(z11-z1)),i;
| | | | | | IF i=-incr then
| | | | | | | PLOT -0.15,i;-0.15,-i;
| | | | | | | PLOT -0.1*(1-(radius2 - (SQR(((a-1)*lambda/2+radius2)^2-j^2))
```

```
| | | | | | | | | -z1)/(z11-z1)),-i
| | | | | | | | PLOT -0.1*(1-(radius2 - (SQR(((a-1)*lambda/2+radius2)^2-j^2))
| | | | | | | | | -z1)/(z11-z1)),i;
| | | | | | | END IF
| | | | | | | EXIT FOR
| | | | | | END IF
| | | | | NEXT a
| | | | NEXT i
| | | END IF
| | FLOOD flood_cent,0
| | |
| | WINDOW #3
| | SET WINDOW 0,incr,0,1
| | SET COLOR 7
| | iii=0.05
| | FOR i=0 to incr STEP incr/10                  !!!
| | | FOR j=-iii/2 to 1+iii/2 STEP 2*iii           !
| | | | PLOT i,j;i,j+iii                           !
| | | NEXT j                                       !
| | NEXT i                                         ! Make
| | iii=incr/20                                    ! grid
| | FOR i=0 to 1.1 STEP 0.1                        ! graphs.
| | | FOR j=-iii/2 to incr+iii/2 STEP 2*iii        !
| | | | PLOT j,i;j+iii,i                           !
| | | NEXT j                                       !
| | NEXT i                                         !
| | SET COLOR 15                                   !
| | PLOT 0,0;incr,0;incr,1;0,1;0,0                 !!!
| | SET TEXT JUSTIFY "center","half"
| | |
| | |          ! For determining the mirror waists of the symmetrical resonator.
| | |
| | IF confined$="Y" then
| | | IF m1$="PLANE" OR m2$="PLANE" OR radius1=radius2 then
| | | | IF radius1=radius2 then
| | | | | cav_length=cavity_length/2
| | | | | waist=SQR(lambda*cav_length/pi)*(2*radius1^2/(cavity_length
| | | | | | *(radius1-cav_length)))^0.25
| | | | | z0=SQR((2*radius1-cavity_length)*cavity_length/4)
| | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | w1=0
| | | | END IF
| | | | IF m1$="PLANE" and m2$="PLANE" then
| | | | | EXIT IF
```

```
| | | | ELSE
| | | | | IF m2$="STEP" then
| | | | | | IF m1$="PLANE" then                          ! w1=w0 ELSE w1=waist
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(8*fzt_focus^2/(cavity_length
| | | | | | | | *(2*fzt_focus-cav_length)))^0.25
| | | | | | | z0=SQR((4*fzt_focus-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | w1=w0
| | | | | | | cavity_length=cavity_length/2
| | | | | | ELSE
| | | | | | | w1=waist
| | | | | | END IF
| | | | | | j=w1/step_max_radius
| | | | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | | ELSE
| | | | | | IF m2$="PLANE" then                        ! w1=waist ELSE w1=w0
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(2*radius1^2/(cavity_length
| | | | | | | | *(radius1-cav_length)))^0.25
| | | | | | | z0=SQR((2*radius1-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | cavity_length=cavity_length/2
| | | | | | | w1=waist
| | | | | | ELSE
| | | | | | | w1=w0
| | | | | | END IF
| | | | | | IF m1$="PLANE" then                          ! w1=w0 ELSE w1=waist
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(2*radius2^2/(cavity_length
| | | | | | | | *(radius2-cav_length)))^0.25
| | | | | | | z0=SQR((2*radius2-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | cavity_length=cavity_length/2
| | | | | | | w1=w0
| | | | | | ELSE
| | | | | | | w1=waist
| | | | | | END IF
| | | | | | j=w1/max_radius1
| | | | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | | END IF
```

```
| | | | | FOR i=0 to incr STEP 2                           !!! Plot gaussian E-Field
| | | | | | PLOT i,EXP(-(i*dr1/w1)^2);                       ! (For symmetrical and
| | | | | | PLOT i+1,EXP(-((i+1)*dr1/w1)^2)                  !  half symmetrical
| | | | | NEXT i                                           !!!  mirrors only).
| | | | | PLOT
| | | | END IF
| | | ELSE                                        ! For non-symmetrical resonators.
| | | | cavity_length_temp=cavity_length-1e-10
| | | | z0_sqrd=cavity_length_temp*(radius1-cavity_length_temp)
| | | | z0_sqrd=z0_sqrd*(radius2-cavity_length_temp)
| | | | z0_sqrd=z0_sqrd*(radius1+radius2-cavity_length_temp)
| | | | z0_sqrd=z0_sqrd/(radius2+radius1-2*cavity_length_temp)^2
| | | | z0=SQR(z0_sqrd)
| | | | w0=SQR(lambda*z0/pi)
| | | | z_m1=-0.5*radius1-0.5*SQR(radius1^2-4*z0_sqrd)
| | | | z_m2= 0.5*radius2+0.5*SQR(radius2^2-4*z0_sqrd)
| | | | w1=w0*SQR(1+(z_m1/z0)^2)
| | | | w2=w0*SQR(1+(z_m2/z0)^2)
| | | | |
| | | | j=w1/max_radius1
| | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | FOR i=0 to incr STEP 2                             !!!
| | | | | PLOT i,EXP(-(i*dr1/w1)^2);                         ! Plot gaussian E-Field
| | | | | PLOT i+1,EXP(-((i+1)*dr1/w1)^2)                    ! (For non-symmetrical
| | | | NEXT i                                             !!!  mirrors only).
| | | | PLOT
| | | END IF
| | END IF
| | SET WINDOW 0,1,0,1
| | BOX CLEAR 0.6,1,0.9,1
| | BOX LINES 0.6,1,0.9,1
| | BOX CLEAR 0.7,0.9,0.8,0.9
| | BOX LINES 0.7,0.9,0.8,0.9
| | PLOT TEXT, AT 0.8,0.95:"FINAL"
| | SET WINDOW 0,incr1,0,1
| | BOX KEEP  0,incr,0,1 IN grid_graph1$
| | |
| | WINDOW #6
| | SET WINDOW 0,incr,0,1
| | SET COLOR 7
| | iii=0.05
| | FOR i=0 to incr STEP incr/10                         !!!
| | | FOR j=-iii/2 to 1+iii/2 STEP 2*iii                   !
| | | | PLOT i,j;i,j+iii                                   !
```

```
| | | NEXT j                                        !
| | NEXT i                                          ! Make
| | iii=incr/20                                     ! grid
| | FOR i=0 to 1.1 STEP 0.1                         ! graphs.
| | | FOR j=-iii/2 to incr+iii/2 STEP 2*iii         !
| | | | PLOT j,i;j+iii,i                            !
| | | NEXT j                                        !
| | NEXT i                                          !
| | SET COLOR 15                                    !
| | PLOT 0,0;incr,0;incr,1;0,1;0,0                !!!
| | SET TEXT JUSTIFY "center","half"
| | |
| | |              ! For determining the mirror waists of the symmetrical resonator.
| | |
| | IF confined$="Y" then
| | | IF m1$="PLANE" OR m2$="PLANE" OR radius1=radius2 then
| | | | IF radius1=radius2 then
| | | | | cav_length=cavity_length/2
| | | | | waist=SQR(lambda*cav_length/pi)*(2*radius1^2/(cavity_length
| | | | | | *(radius1-cav_length)))^0.25
| | | | | z0=SQR((2*radius1-cavity_length)*cavity_length/4)
| | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | w2=0
| | | | END IF
| | | | IF m1$="PLANE" and m2$="PLANE" then
| | | | | EXIT IF
| | | | ELSE
| | | | | IF m1$="STEP" then
| | | | | | IF m2$="PLANE" then                       ! w2=w0 ELSE w2=waist
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(8*fzt_focus^2/(cavity_length
| | | | | | | | *(2*fzt_focus-cav_length)))^0.25
| | | | | | | z0=SQR((4*fzt_focus-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | w2=w0
| | | | | | | cavity_length=cavity_length/2
| | | | | | ELSE
| | | | | | | w2=waist
| | | | | | END IF
| | | | | | j=w2/step_max_radius
| | | | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | | ELSE
| | | | | | IF m1$="PLANE" then                       ! w2=waist ELSE w2=w0
```

```
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(2*radius2^2/(cavity_length
| | | | | | | | *(radius2-cav_length)))^0.25
| | | | | | | z0=SQR((2*radius2-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | cavity_length=cavity_length/2
| | | | | | | w2=waist
| | | | | | ELSE
| | | | | | | w2=w0
| | | | | | END IF
| | | | | | IF m2$="PLANE" then                        ! w2=w0 ELSE w2=waist
| | | | | | | cavity_length=cavity_length*2
| | | | | | | cav_length=cavity_length/2
| | | | | | | waist=SQR(lambda*cav_length/pi)*(2*radius1^2/(cavity_length
| | | | | | | | *(radius1-cav_length)))^0.25
| | | | | | | z0=SQR((2*radius1-cavity_length)*cavity_length/4)
| | | | | | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | | | | | cavity_length=cavity_length/2
| | | | | | | w2=w0
| | | | | | ELSE
| | | | | | | w2=waist
| | | | | | END IF
| | | | | | j=w2/max_radius2
| | | | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | | END IF
| | | | | FOR i=0 to incr STEP 2                    !!! Plot gaussian E-Field
| | | | | | PLOT i,EXP(-(i*dr2/w2)^2);               ! (for symmetrical and
| | | | | | PLOT i+1,EXP(-((i+1)*dr2/w2)^2)          !  half symmetrical
| | | | | NEXT i                                  !!!  mirrors only).
| | | | | PLOT
| | | | END IF
| | | ELSE                                        ! For non-symmetrical resonators.
| | | | j=w2/max_radius2
| | | | IF j<1 then PLOT TEXT, AT incr*j,EXP(-1):"*"
| | | | FOR i=0 to incr STEP 2                     !!!
| | | | | PLOT i,EXP(-(i*dr2/w2)^2);                ! Plot gaussian E-Field
| | | | | PLOT i+1,EXP(-((i+1)*dr2/w2)^2)           ! (For non-symmetrical
| | | | NEXT i                                   !!!  mirrors only).
| | | | PLOT
| | | END IF
| | END IF
| | |
| | SET WINDOW 0,1,0,1
```

```
| | BOX CLEAR 0.6,1,0.9,1
| | BOX LINES 0.6,1,0.9,1
| | BOX CLEAR 0.7,0.9,0.8,0.9
| | BOX LINES 0.7,0.9,0.8,0.9
| | PLOT TEXT, AT 0.8,0.95:"FINAL"
| | SET WINDOW 0,incr,0,1
| | BOX KEEP  0,incr,0,1 IN grid_graph2$
| | FOR i=3 to 6 STEP 3
| | | WINDOW #i
| | | SET WINDOW 0,1,0,1
| | | SET COLOR 15
| | | SET TEXT JUSTIFY "center","half"
| | | BOX CLEAR 0.4,1,0.9,1
| | | BOX LINES 0.4,1,0.9,1
| | | PLOT TEXT, AT 0.7,0.95:"HISTORY"
| | | BOX CLEAR 0.3,1,0.8,0.9
| | | BOX LINES 0.3,1,0.8,0.9
| | | PLOT TEXT, AT 0.65,0.85:"EVERY 10"
| | | IF i<5 then SET WINDOW 0,max_radius1,0,1 ELSE SET WINDOW
| | | | 0,max_radius2,0,1
| | NEXT i
| | |
| | CALL labels
| | WINDOW #1
| | SET WINDOW 0.05,0.95,0,1
| | SET COLOR 15
| | SET TEXT JUSTIFY "left","half"
| | PLOT TEXT, AT 0.05,0.425:" M1"
| | SET TEXT JUSTIFY "right","half"
| | PLOT TEXT, AT 0.95,0.425:"M2 "
| | SET TEXT JUSTIFY "center","half"
| | PLOT TEXT, AT 0.1927,0.425:"Norm Amp vs R"
| | PLOT TEXT, AT 0.4880,0.425:"Norm Amp,phase vs R"
| | PLOT TEXT, AT 0.8074,0.425:"Norm Amp vs R"
| | PLOT TEXT, AT 0.2933,0.80:"1"
| | PLOT TEXT, AT 0.2933,0.45:"0"
| | PLOT TEXT, AT 0.6947,0.80:"+pi "
| | PLOT TEXT, AT 0.6947,0.45+0.035*8:" 0  "
| | PLOT TEXT, AT 0.6947,0.45+0.035*6:"-pi "
| | PLOT TEXT, AT 0.6947,0.45+0.035*4:"-2pi"
| | PLOT TEXT, AT 0.6947,0.45+0.035*2:"-3pi"
| | PLOT TEXT, AT 0.6947,0.45:"-4pi"
| | |
| | SET CURSOR 2,1                                    ! Header section.
```

```
| | PRINT USING "Incr=###; Inct=###":incr,inct
| | |
| | SET CURSOR 5,1
| | IF m1$="TIERED" then
| | | PRINT "M1=";m1$;"(";STR$(tpz1);")"
| | ELSE
| | | PRINT "M1=";m1$
| | END IF
| | |
| | SET CURSOR 5,51
| | IF m2$="TIERED" then
| | | PRINT "M2=";m2$;"(";STR$(tpz2);")"
| | ELSE
| | | PRINT "M2=";m2$
| | END IF
| | IF m1$="STEP" and m2$="STEP" then
| | | SET CURSOR 5,51
| | | PRINT "M2=STEP; ALTERNATE=";alternate$
| | END IF
| | |
| | SET CURSOR 1,51
| | PRINT "R1=";
| | IF m1$="PLANE" OR m1$="PARABOLIC" then
| | | PRINT m1$[1:5];"; R2=";
| | ELSE
| | | IF radius1>0 and radius1<1 then
| | | | PRINT "0";STR$(radius1);"; R2=";
| | | ELSEIF radius1>-1 and radius1<0 then
| | | | PRINT "-0";STR$(-radius1);"; R2=";
| | | ELSE
| | | | PRINT STR$(radius1);"; R2=";
| | | END IF
| | END IF
| | |
| | IF m2$="PLANE" OR m2$="PARABOLIC" then
| | | PRINT m2$[1:5]
| | ELSE
| | | IF radius2>0 and radius2<1 then
| | | | PRINT "0";STR$(radius2)
| | | ELSEIF radius2>-1 and radius2<0 then
| | | | PRINT "-0";STR$(-radius2)
| | | ELSE
| | | | PRINT STR$(radius2)
| | | END IF
```

```
1030  | | END IF
1031  | | |
1032  | | SET CURSOR 2,51
1033  | | PRINT "N1=";
1034  | | n_number1=max_radius1^2/cavity_length/lambda
1035  | | IF n_number1<1 then
1036  | | | PRINT "0";STR$(n_number1)[1:4];"; N2=";
1037  | | ELSE
1038  | | | PRINT STR$(n_number1)[1:5];"; N2=";
1039  | | END IF
1040  | | |
1041  | | n_number2=max_radius2^2/cavity_length/lambda
1042  | | IF n_number2<1 then
1043  | | | PRINT "0";STR$(n_number2)[1:4]
1044  | | ELSE
1045  | | | PRINT STR$(n_number2)[1:5]
1046  | | END IF
1047  | | |
1048  | | SET CURSOR 3,1
1049  | | PRINT "W=Wavelength=";STR$(lambda/1e-6);" um"
1050  | | |
1051  | | SET CURSOR 4,51
1052  | | PRINT "Input Wave=";input_wave$[1:5]
1053  | | |
1054  | | IF m1$="STEP" then
1055  | | | n_number=step_max_radius^2/cavity_length/lambda
1056  | | | SET CURSOR 4,1
1057  | | | PRINT "Etch Depth=";etch_depth$
1058  | | | SET CURSOR 3,51
1059  | | | PRINT "# of Zones=";STR$(step_tiers);"; N=";
1060  | | | IF n_number<1 then
1061  | | | | PRINT "0";STR$(n_number)[1:4]
1062  | | | ELSE
1063  | | | | PRINT STR$(n_number)[1:5]
1064  | | | END IF
1065  | | ELSEIF m1$="TIERED" and m2$="TIERED" then
1066  | | | SET CURSOR 4,1
1067  | | | PRINT "Etch Depth1,2=W/(";STR$(2*tpz1);",";STR$(2*tpz2);")"
1068  | | ELSEIF m1$="TIERED" then
1069  | | | SET CURSOR 4,1
1070  | | | PRINT "Etch Depth1=";etch_depth1$
1071  | | ELSEIF m2$="TIERED" then
1072  | | | SET CURSOR 4,1
1073  | | | PRINT "Etch Depth2=";etch_depth2$
```

```
| | END IF
| | |
| | SET CURSOR 3,51
| | IF m1$="TIERED" and m2$="TIERED" then
| | | PRINT "tiers1,2=";STR$(tiers1);",";STR$(tiers2);"; d=";
| | ELSEIF m1$="TIERED" then
| | | PRINT "tiers1=";STR$(tiers1);"; d=";
| | ELSEIF m2$="TIERED" then
| | | PRINT "tiers2=";STR$(tiers2);"; d=";
| | ELSE
| | | PRINT "d=";
| | END IF
| | IF cavity_length<1 then PRINT "0";
| | PRINT STR$(cavity_length)
| | |
| | SET CURSOR 5,24
| | PRINT USING "Begin:######## ##/##/##":&
| | &          time$,date$[5:6],date$[7:8],date$[3:4]
| | SET CURSOR 1,24
| | PRINT "Prgm = RESONATE ver. 1.0"
| | SET CURSOR 4,1
| | IF similate$="Y" then PRINT sim$;";";tol*1e10
| | |
| | |                                               ! Define the input wave.
| | |
| | IF input_wave$="PLANE" then
| | | MAT re2=1                          ! Initialize all the elements of re2 & im2.
| | | MAT im2=0                          !  (re = Real part & im = Imaginary part)
| | ELSE
| | | cav_length=cavity_length/2
| | | IF m1$<>"PLANE" then
| | | | z0=SQR((2*radius1-cavity_length)*cavity_length/4)
| | | ELSE
| | | | IF m2$<>"PLANE" then z0=SQR((2*radius2-cavity_length)
| | | | | *cavity_length/4)
| | | END IF
| | | w0=waist/SQR(1+(cav_length/z0)^2)
| | | eta=ATN(cav_length/z0)
| | | Rz=cav_length*(1+(z0/cav_length)^2)
| | | FOR i=0 to incr                               ! Gaussian input wave.
| | | | re2(i+1)=2*w0/waist*EXP(-(i*dr1/waist)^2)*cos(eta-k*(cav_length
| | | | | +(i*dr1)^2/2/Rz))                        ! Real part.
| | | NEXT i
| | | FOR i=0 to incr                               ! Imaginary part.
```

```
| | | | im2(i+1)=2*w0/waist*EXP(-(i*dr1/waist)^2)*sin(eta-k*(cav_length
| | | | | +(i*dr1)^2/2/Rz))                                   ! Real part.
| | | NEXT i
| | END IF
| | |
| | max_amp=-1                                              !!!
| | FOR i=1 to incr+1                                        !
| | | amp=SQR(re2(i)^2+im2(i)^2)                             ! Find the total
| | | zzm(i)= amp                                            ! initial
| | | max_amp=max(max_amp,amp)                               ! amplitude
| | NEXT i                                                   ! distribution
| | MAT zzm=(1/max_amp)*zzm                                  ! incident on
| | CALL Simpsons_Rule1(zzm,1,incr+1,initial_int1_amp)       ! Mirror1.
| | initial_int1_amp=initial_int1_amp/incr                   !
| | last_int1_amp=initial_int1_amp                          !!!
| | |
| | FOR i=1 to incr+1                                       !!!
| | | j=(i-1)*dr1                                            ! Find the total
| | | zzm(i)=(re2(i)^2+im2(i)^2)*j                           ! initial power
| | NEXT i                                                   ! incident on
| | CALL Simpsons_Rule1(zzm,dr1,incr+1,initial_power1)       ! mirror1.
| | initial_power1=initial_power1*2*pi                       !
| | last_power1=initial_power1                               !
| | last_power=initial_power1                               !!!
| | |
| | WINDOW #15
| | SET WINDOW 0.05,0.95,0,1
| | color=8
| | phase=1                      ! 1=Reference plane or 0=mirror surface.
| | rt=0                         ! Initialize the round trip count.
| | |
| | |                            ! The main body of the prgm begins here.
| | |
| | FOR transits=1 to max_transits          ! Number of transits or passes.
| | | MAT re1=re2                           ! (2 transits = 1 round trip).
| | | MAT im1=im2
| | | mirror=1+MOD(transits,2)              ! Mirror tracker.
| | | WINDOW #1
| | | SET CURSOR 1,1
| | | PRINT "            ";
| | | SET CURSOR 1,1
| | | SET COLOR 15
| | | IF mirror=1 then
| | | | PRINT "(";
```

```
| | | | SET COLOR 11
| | | | PRINT "<-";
| | | | SET COLOR 10
| | | | PRINT "--";
| | | | SET COLOR 15
| | | | PRINT USING ") RT= ### of <####":rt,STR$(max_rt)
| | | | WINDOW #14
| | | | SET WINDOW 0,incr+3,0,2
| | | | SET TEXT JUSTIFY "center","half"
| | | | CLEAR
| | | | SET COLOR 11
| | | | MAT phase_delay=(phase*k)*zz1
| | | ELSEIF mirror=2 then
| | | | rt=rt+1                                ! Round trip count. The initial
| | | | PRINT "(";                             ! wave originates from Mirror2.
| | | | SET COLOR 11
| | | | PRINT "--";
| | | | SET COLOR 10
| | | | PRINT "->";
| | | | SET COLOR 15
| | | | PRINT USING ") RT= ### of <####":rt-1,STR$(max_rt)
| | | | WINDOW #14
| | | | SET WINDOW 0,incr+3,0,2
| | | | SET TEXT JUSTIFY "center","half"
| | | | CLEAR
| | | | SET COLOR 10
| | | | MAT phase_delay=(phase*k)*zz2
| | | END IF
| | | IF MOD(transits-1,10)=0 then color=color+1            ! MOD=remainder.
| | | IF MOD(color,16)=0 then color=9
| | | |
| | | IF transits<3 then
| | | | IF transits=1 then                        ! mirror=2
| | | | | FOR n2=1 to incr+1                      ! Find the resultant E field
| | | | | | r2=(n2-1)*dr2                         ! on Mirror2 due to Mirror1.
| | | | | | r2r2=r2*r2
| | | | | | z2=cavity_length-zz2(n2)
| | | | | | |
| | | | | | IF show$="Y" then PLOT TEXT, AT n2,1:">"
| | | | | | |
| | | | | | FOR n1=2 to incr+1                    ! Integrate over Mirror1.
| | | | | | | z,z_1(n1)=z2-zz1(n1)
| | | | | | | r1=(n1-1)*dr1
| | | | | | | zr1r2,zr1r2_1(n2,n1)=z*z + r1*r1 + r2r2
```

```
| | | | | | | r1r22,r1r22_1(n2,n1)=r1*r2*2
| | | | | | | r1_lambda,r1_lambda_1(n1)=r1/lambda
| | | | | | | re1_n1=re1(n1)
| | | | | | | im1_n1=im1(n1)
| | | | | | | FOR t1=1 to inct+1
| | | | | | | | r=SQR(zr1r2 - r1r22*cos_t(t1))
| | | | | | | | kr=k*r
| | | | | | | | h=(1+z/r)*r1_lambda/r
| | | | | | | | sin_kr=sin(kr)
| | | | | | | | cos_kr=cos(kr)
| | | | | | | | gre(t1)=h*(re1_n1*sin_kr - im1_n1*cos_kr)
| | | | | | | | gim(t1)=h*(re1_n1*cos_kr + im1_n1*sin_kr)
| | | | | | | NEXT t1
| | | | | | | CALL Simpsons_Rule2(gre,gim,dt,inct+1,gre2(n1),gim2(n1))
| | | | | | NEXT n1
| | | | | | CALL Simpsons_Rule2(gre2,gim2,dr1,incr+1,re2(n2),im2(n2))
| | | | | NEXT n2
| | | | ELSEIF transits=2 then                    ! Mirror=1
| | | | | FOR n1=1 to incr+1                      ! Find the resultant E field
| | | | | | r1=(n1-1)*dr1                         ! on Mirror1 due to Mirror2.
| | | | | | r1r1=r1*r1
| | | | | | z1=cavity_length-zz1(n1)
| | | | | | |
| | | | | | IF show$="Y" then PLOT TEXT, AT incr+2-n1,1:"<"
| | | | | | |
| | | | | | FOR n2=2 to incr+1                    ! Integrate over Mirror2.
| | | | | | | z,z_2(n2)=z1-zz2(n2)
| | | | | | | r2=(n2-1)*dr2
| | | | | | | zr1r2,zr1r2_2(n1,n2)=z*z + r2*r2 + r1r1
| | | | | | | r1r22,r1r22_2(n1,n2)=r1*r2*2
| | | | | | | r2_lambda,r2_lambda_2(n2)=r2/lambda
| | | | | | | re1_n2=re1(n2)
| | | | | | | im1_n2=im1(n2)
| | | | | | | FOR t2=1 to inct+1
| | | | | | | | r=SQR(zr1r2 - r1r22*cos_t(t2))
| | | | | | | | kr=k*r
| | | | | | | | h=(1+z/r)*r2_lambda/r
| | | | | | | | sin_kr=sin(kr)
| | | | | | | | cos_kr=cos(kr)
| | | | | | | | gre(t2)=h*(re1_n2*sin_kr - im1_n2*cos_kr)
| | | | | | | | gim(t2)=h*(re1_n2*cos_kr + im1_n2*sin_kr)
| | | | | | | NEXT t2
| | | | | | | CALL Simpsons_Rule2(gre,gim,dt,inct+1,gre2(n2),gim2(n2))
| | | | | | NEXT n2
```

```
| | | | | | CALL Simpsons_Rule2(gre2,gim2,dr2,incr+1,re2(n1),im2(n1))
| | | | | NEXT n1
| | | | END IF
| | | ELSE
| | | | IF mirror=2 then
| | | | | FOR n2=1 to incr+1                    ! Find the resultant E field
| | | | | | |                                  ! on Mirror2 due to Mirror1.
| | | | | | |
| | | | | | IF show$="Y" then PLOT TEXT, AT n2,1:">"
| | | | | | |
| | | | | | FOR n1=2 to incr+1                 ! Integrate over Mirror1.
| | | | | | | zr1r2=zr1r2_1(n2,n1)
| | | | | | | r1r22=r1r22_1(n2,n1)
| | | | | | | z=z_1(n1)
| | | | | | | r1_lambda=r1_lambda_1(n1)
| | | | | | | re1_n1=re1(n1)
| | | | | | | im1_n1=im1(n1)
| | | | | | | FOR t1=1 to inct+1
| | | | | | | | r=SQR(zr1r2 - r1r22*cos_t(t1))
| | | | | | | | kr=k*r
| | | | | | | | h=(1+z/r)*r1_lambda/r
| | | | | | | | sin_kr=sin(kr)
| | | | | | | | cos_kr=cos(kr)
| | | | | | | | gre(t1)=h*(re1_n1*sin_kr - im1_n1*cos_kr)
| | | | | | | | gim(t1)=h*(re1_n1*cos_kr + im1_n1*sin_kr)
| | | | | | | NEXT t1
| | | | | | | CALL Simpsons_Rule2(gre,gim,dt,inct+1,gre2(n1),gim2(n1))
| | | | | | NEXT n1
| | | | | | CALL Simpsons_Rule2(gre2,gim2,dr1,incr+1,re2(n2),im2(n2))
| | | | | NEXT n2
| | | | ELSEIF mirror=1 then
| | | | | FOR n1=1 to incr+1                    ! Find the resultant E field
| | | | | | |                                  ! on Mirror1 due to Mirror2.
| | | | | | |
| | | | | | IF show$="Y" then PLOT TEXT, AT incr+2-n1,1:"<"
| | | | | | |
| | | | | | FOR n2=2 to incr+1                 ! Integrate over Mirror2.
| | | | | | | zr1r2=zr1r2_2(n1,n2)
| | | | | | | r1r22=r1r22_2(n1,n2)
| | | | | | | z=z_2(n2)
| | | | | | | r2_lambda=r2_lambda_2(n2)
| | | | | | | re1_n2=re1(n2)
| | | | | | | im1_n2=im1(n2)
| | | | | | | FOR t2=1 to inct+1
```

```
| | | | | | | | | r=SQR(zr1r2 - r1r22*cos_t(t2))
| | | | | | | | | kr=k*r
| | | | | | | | | h=(1+z/r)*r2_lambda/r
| | | | | | | | | sin_kr=sin(kr)
| | | | | | | | | cos_kr=cos(kr)
| | | | | | | | | gre(t2)=h*(re1_n2*sin_kr - im1_n2*cos_kr)
| | | | | | | | | gim(t2)=h*(re1_n2*cos_kr + im1_n2*sin_kr)
| | | | | | | | NEXT t2
| | | | | | | | CALL Simpsons_Rule2(gre,gim,dt,inct+1,gre2(n2),gim2(n2))
| | | | | | | NEXT n2
| | | | | | | CALL Simpsons_Rule2(gre2,gim2,dr2,incr+1,re2(n1),im2(n1))
| | | | | | NEXT n1
| | | | | END IF
| | | | END IF
| | | |
| | | WINDOW #1
| | | |
| | | IF transits=1 then
| | | | max_amp=-1                                            !!!
| | | | FOR i=1 to incr+1                                      !
| | | | | amp=SQR(re2(i)^2+im2(i)^2)                           !
| | | | | zzm(i)= amp                                          ! Find the
| | | | | max_amp=max(max_amp,amp)                             ! total initial
| | | | | IF max_amp=amp then max_amp_i=i                      ! distribution
| | | | NEXT i                                                 ! incident on
| | | | MAT zzm=(1/max_amp)*zzm                                ! mirror2.
| | | | CALL Simpsons_Rule1(zzm,1,incr+1,initial_int2_amp)     !
| | | | initial_int2_amp=initial_int2_amp/incr                 !
| | | | last_int2_amp=initial_int2_amp                        !!!
| | | | |
| | | ELSEIF mirror=1 then
| | | | max_amp=-1                                            !!!
| | | | FOR i=1 to incr+1                                      !
| | | | | amp=SQR(re2(i)^2+im2(i)^2)                           ! Find the
| | | | | zzm(i)= amp                                          ! maximum
| | | | | max_amp=max(max_amp,amp)                             ! amplitude and
| | | | | IF max_amp=amp then max_amp_i=i                      ! distribution
| | | | NEXT i                                                 ! incident on
| | | | MAT zzm=(1/max_amp)*zzm                                ! mirror1.
| | | | CALL Simpsons_Rule1(zzm,1,incr+1,int1_amp)             !
| | | | int1_amp=int1_amp/incr                                !!!
| | | | |
| | | ELSEIF mirror=2 then
| | | | max_amp=-1                                            !!!
```

```
| | | | FOR i=1 to incr+1                                         !
| | | | | amp=SQR(re2(i)^2+im2(i)^2)                              ! Find the
| | | | | zzm(i)= amp                                             ! maximum
| | | | | max_amp=max(max_amp,amp)                                ! amplitude and
| | | | | IF max_amp=amp then max_amp_i=i                         ! distribution
| | | | NEXT i                                                    ! incident on
| | | | MAT zzm=(1/max_amp)*zzm                                   ! mirror2.
| | | | CALL Simpsons_Rule1(zzm,1,incr+1,int2_amp)                !
| | | | int2_amp=int2_amp/incr                                  !!!
| | | END IF
| | | |
| | | IF mirror=1 then
| | | | IF transits>1 then last_last_dev_amp=last_dev_amp
| | | | dev_amp=100*(1-(int1_amp+past_int1_amp)/(2*int1_amp))
| | | | SET CURSOR 3,24
| | | | PRINT "                    "
| | | | SET CURSOR 3,24
| | | | PRINT USING "Dev Amp   =+#.####^^^^ #":dev_amp,"%"
| | | | past_int1_amp=int1_amp
| | | | last_dev_amp=dev_amp
| | | | IF transits>3 and ABS(dev_amp)<dev and ABS(last_dev_amp)<dev and
| | | | | ABS(last_last_dev_amp)<dev then
| | | | | |
| | | | | |                                   ! Stop if the last 3 transits are
| | | | | |                                   ! within dev (the fluxuation factor).
| | | | | |
| | | | | WINDOW #4
| | | | | SET WINDOW 0,1,0,1
| | | | | SET TEXT JUSTIFY "center","half"
| | | | | SET COLOR 15
| | | | | BOX CLEAR 0.6,1,0.8,0.9
| | | | | BOX LINES 0.6,1,0.8,0.9
| | | | | PLOT TEXT, AT 0.8,0.85:"STABLE"
| | | | | WINDOW #5
| | | | | SET WINDOW 0,1,0,1
| | | | | SET TEXT JUSTIFY "center","half"
| | | | | SET COLOR 15
| | | | | BOX CLEAR 0.6,1,0.8,0.9
| | | | | BOX LINES 0.6,1,0.8,0.9
| | | | | PLOT TEXT, AT 0.8,0.85:"STABLE"
| | | | | WINDOW #14
| | | | | SET WINDOW 0,incr+3,0,2
| | | | | SET TEXT JUSTIFY "center","half"
| | | | | CLEAR
```

```
| | | | | STOP
| | | | END IF
| | | END IF
| | | SET CURSOR 4,24
| | | PRINT USING " End:######## ##/##/##":&
| | | &          time$,date$[5:6],date$[7:8],date$[3:4]
| | | |
| | | |
| | | IF MOD(transits-1,10)=0 then               ! Every 10th transit (1,11,21,...).
| | | | WINDOW #6                                                   ! mirror=2.
| | | | SET COLOR color
| | | | FOR i=1 to incr+1                                    !!! Plot the
| | | | | PLOT (i-1)*dr2,SQR(re2(i)^2+im2(i)^2)/max_amp;      ! relative
| | | | NEXT i                                               !!! amplitudes.
| | | | PLOT
| | | | IF w2<>0 then
| | | | | SET COLOR 15
| | | | | SET TEXT JUSTIFY "center","half"
| | | | | IF m1$="STEP" then
| | | | | | j=w2/step_max_radius2
| | | | | | IF j<1 then PLOT TEXT, AT (incr*j)*dr2,EXP(-1):"*"
| | | | | ELSE
| | | | | | j=w2/max_radius2
| | | | | | IF j<1 then PLOT TEXT, AT (incr*j)*dr2,EXP(-1):"*"
| | | | | END IF
| | | | | FOR i=0 to incr STEP 2                           !!!
| | | | | | PLOT i*dr2,EXP(-(i*dr2/w2)^2);                 ! Plot the theoretical
| | | | | | PLOT (i+1)*dr2,EXP(-((i+1)*dr2/w2)^2)          ! gaussian E-Field.
| | | | | NEXT i                                           !!!
| | | | | PLOT
| | | | | SET COLOR color
| | | | END IF
| | | END IF
| | | |
| | | IF MOD(transits-2,10)=0 then               ! Every 10th transit (2,12,22,...).
| | | | WINDOW #3                                                   ! mirror=1.
| | | | SET COLOR color
| | | | FOR i=1 to incr+1                                !!! Plot the
| | | | | PLOT (i-1)*dr1,SQR(re2(i)^2+im2(i)^2)/max_amp; ! relative
| | | | NEXT i                                           !!! amplitudes.
| | | | PLOT
| | | | IF w1<>0 then
| | | | | SET COLOR 15
| | | | | SET TEXT JUSTIFY "center","half"
```

```
| | | | | IF m1$="STEP" then
| | | | | | j=w1/step_max_radius1
| | | | | | IF j<1 then PLOT TEXT, AT (incr*j)*dr1,EXP(-1):"*"
| | | | | ELSE
| | | | | | j=w1/max_radius1
| | | | | | IF j<1 then PLOT TEXT, AT (incr*j)*dr1,EXP(-1):"*"
| | | | | END IF
| | | | | FOR i=0 to incr STEP 2                        !!!
| | | | | | PLOT i*dr1,EXP(-(i*dr1/w1)^2);               ! Plot the theoretical
| | | | | | PLOT (i+1)*dr1,EXP(-((i+1)*dr1/w1)^2)        ! gaussian E-Field.
| | | | | NEXT i                                        !!!
| | | | | PLOT
| | | | | SET COLOR color
| | | | END IF
| | | END IF
| | | |
| | | IF mirror=1 then
| | | | WINDOW #4
| | | | SET WINDOW 0,1,0,1
| | | | CLEAR
| | | | SET COLOR 15
| | | | SET TEXT JUSTIFY "center","half"
| | | | BOX SHOW grid_graph1$ AT 0,0 USING "OR"
| | | | PLOT TEXT, AT 0.8,0.85:"M1"
| | | | SET COLOR 10                                   ! intensified "green"
| | | | SET WINDOW 0,max_radius1,0,1
| | | | FOR i=1 to incr+1                              !!!  Plot the
| | | | | PLOT (i-1)*dr1,SQR(re2(i)^2+im2(i)^2)/max_amp;  !  relative
| | | | NEXT i                                         !!!  amplitudes.
| | | | PLOT
| | | | |
| | | | |                                         ! Plot relative phase below.
| | | | |
| | | | SET WINDOW 0,max_radius1,-4*pi,pi
| | | | SET COLOR "yellow"
| | | | IF re2(max_amp_i)>0 then
| | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)
| | | | ELSE
| | | | | IF im2(max_amp_i)>0 then
| | | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)+pi
| | | | | ELSE
| | | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)-pi
| | | | | END IF
| | | | END IF
```

```
| | | | FOR ii=1 to incr-1 STEP 2
| | | | | FOR i=ii to ii+1
| | | | | | IF re2(i)>0 then
| | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)
| | | | | | ELSE
| | | | | | | IF im2(i)>0 then
| | | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)+pi
| | | | | | | ELSE
| | | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)-pi
| | | | | | | END IF
| | | | | | END IF
| | | | | | j=jjj-jj-phase_delay(i)+phase_delay(max_amp_i)
| | | | | | IF i>1 then
| | | | | | | DO UNTIL(ABS(last_j-j)<1.5*pi)
| | | | | | | | IF j>last_j then
| | | | | | | | | j=j-2*pi
| | | | | | | | ELSE
| | | | | | | | | j=j+2*pi
| | | | | | | | END IF
| | | | | | | LOOP
| | | | | | END IF
| | | | | | last_j=j
| | | | | | PLOT (i-1)*dr1,j;
| | | | | NEXT i
| | | | | PLOT
| | | | NEXT ii
| | | | PLOT
| | | ELSEIF mirror=2 then
| | | | WINDOW #5
| | | | SET WINDOW 0,1,0,1
| | | | CLEAR
| | | | SET COLOR 15
| | | | SET TEXT JUSTIFY "center","half"
| | | | BOX SHOW grid_graph2$ AT 0,0 USING "OR"
| | | | PLOT TEXT, AT 0.8,0.85:"M2"
| | | | SET COLOR 11                                        ! intensified "cyan"
| | | | SET WINDOW 0,max_radius2,0,1
| | | | FOR i=1 to incr+1                                   !!! Plot the
| | | | | PLOT (i-1)*dr2,SQR(re2(i)^2+im2(i)^2)/max_amp;  ! relative
| | | | NEXT i                                              !!!  amplitudes.
| | | | PLOT
| | | | |
| | | | |                                               ! Plot relative phase below.
| | | | |
```

```
| | | | SET WINDOW 0,max_radius2,-4*pi,pi
| | | | SET COLOR "yellow"
| | | | IF re2(max_amp_i)>0 then
| | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)
| | | | ELSE
| | | | | IF im2(max_amp_i)>0 then
| | | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)+pi
| | | | | ELSE
| | | | | | jj=ATN(im2(max_amp_i)/re2(max_amp_i)+1e-10)-pi
| | | | | END IF
| | | | END IF
| | | | FOR ii=1 to incr-1 STEP 2
| | | | | FOR i=ii to ii+1
| | | | | | IF re2(i)>0 then
| | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)
| | | | | | ELSE
| | | | | | | IF im2(i)>0 then
| | | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)+pi
| | | | | | | ELSE
| | | | | | | | jjj=ATN(im2(i)/re2(i)+1e-10)-pi
| | | | | | | END IF
| | | | | | END IF
| | | | | | j=jjj-jj-phase_delay(i)+phase_delay(max_amp_i)
| | | | | | IF i>1 then
| | | | | | | DO UNTIL(ABS(last_j-j)<1.5*pi)
| | | | | | | | IF j>last_j then
| | | | | | | | | j=j-2*pi
| | | | | | | | ELSE
| | | | | | | | | j=j+2*pi
| | | | | | | | END IF
| | | | | | | LOOP
| | | | | | END IF
| | | | | | last_j=j
| | | | | | PLOT (i-1)*dr2,j;
| | | | | NEXT i
| | | | | PLOT
| | | | NEXT ii
| | | | PLOT
| | | END IF
| | | |
| | | IF transits=1 then
| | | | FOR i=1 to incr+1                                  !!!
| | | | | j=(i-1)*dr2                                       !
| | | | | zzm(i)=(re2(i)^2+im2(i)^2)*j                      ! Find the
```

```
| | | | NEXT i                                                    ! total power
| | | | CALL Simpsons_Rule1(zzm,dr2,incr+1,initial_power2) ! incident
| | | | initial_power2=initial_power2*2*pi                        ! on Mirror2.
| | | | last_power2=initial_power2                                !
| | | | power,power2=initial_power2                              !!!
| | | | |
| | | ELSEIF mirror=1 then
| | | | FOR i=1 to incr+1                                        !!!
| | | | | j=(i-1)*dr1                                             ! Find the
| | | | | zzm(i)=(re2(i)^2+im2(i)^2)*j                            ! total power
| | | | NEXT i                                                    ! incident on
| | | | CALL Simpsons_Rule1(zzm,dr1,incr+1,power1)                ! mirror1.
| | | | power1=power1*2*pi                                        !
| | | | power=power1                                             !!!
| | | | |
| | | ELSEIF mirror=2 then
| | | | FOR i=1 to incr+1                                        !!!
| | | | | j=(i-1)*dr2                                             ! Find the
| | | | | zzm(i)=(re2(i)^2+im2(i)^2)*j                            ! total power
| | | | NEXT i                                                    ! incident on
| | | | CALL Simpsons_Rule1(zzm,dr2,incr+1,power2)                ! Mirror2.
| | | | power2=power2*2*pi                                        !
| | | | power=power2                                             !!!
| | | END IF
| | | |
| | | |                  ! NORmalized POWER is the amount remaining of the initial.
| | | |
| | | WINDOW #1
| | | SET COLOR 15
| | | SET CURSOR 2,24
| | | PRINT "                    "
| | | SET CURSOR 2,24
| | | PRINT USING "Norm Power=+#.####^^^^ #":100*power/initial_power1,
| | | | "%"
| | | |
| | | |                                  ! Plot Power Loss (P.L.) per pass below.
| | | |
| | | IF transits=1 then                                      ! Mirror2 info only.
| | | | loss_per_pass(1)=1-initial_power2/initial_power1
| | | | last_power=power
| | | | loss2_per_pass(1)=loss_per_pass(1)
| | | | last_power2=power2
| | | | loss_min,loss_min_old,loss_min1=1e10
| | | | loss_max,loss_max_old,loss_max1=-1e10
```

```
| | | | IF auto_scale$="Y" then
| | | | | FOR i=ROUND(LOG10(loss2_per_pass(1)))-1 to 0
| | | | | | IF loss2_per_pass(1)>10^i then
| | | | | | | EXIT IF
| | | | | | ELSE
| | | | | | | loss_max2=10^i
| | | | | | | loss_min2=10^(i-1)
| | | | | | | i=0
| | | | | | END IF
| | | | | NEXT i
| | | | ELSE
| | | | | loss_min2=0.00001
| | | | | loss_max2=1
| | | | END IF
| | | | |
| | | ELSEIF mirror=2 then                                ! Mirror2 info only.
| | | | loss2_per_pass(rt)=1-power2/last_power1
| | | | last_power2=power2
| | | | loss_per_pass(transits)=1-power/last_power
| | | | last_power=power
| | | | IF auto_scale$="Y" then
| | | | | FOR i=ROUND(LOG10(loss2_per_pass(rt)))-1 to 0
| | | | | | IF loss2_per_pass(rt)>10^i then
| | | | | | | EXIT IF
| | | | | | ELSE
| | | | | | | loss_max2=10^i
| | | | | | | loss_min2=10^(i-1)
| | | | | | | i=0
| | | | | | END IF
| | | | | NEXT i
| | | | ELSE
| | | | | loss_min2=0.00001
| | | | | loss_max2=1
| | | | END IF
| | | | |
| | | ELSEIF transits=2 then                              ! Mirror1 info only.
| | | | loss1_per_pass(1)=1-power1/last_power2
| | | | last_power1=power1
| | | | loss_per_pass(transits)=1-power/last_power
| | | | last_power=power
| | | | IF auto_scale$="Y" then
| | | | | FOR i=ROUND(LOG10(loss1_per_pass(1)))-1 to 0
| | | | | | IF loss1_per_pass(1)>10^i then
| | | | | | | EXIT IF
```

```
| | | | | | ELSE
| | | | | | | loss_max1=10^i
| | | | | | | loss_min1=10^(i-1)
| | | | | | | i=0
| | | | | | END IF
| | | | | NEXT i
| | | | ELSE
| | | | | loss_min1=0.00001
| | | | | loss_max1=1
| | | | END IF
| | | | |
| | | ELSEIF mirror=1 then                                  ! Mirror1 info only.
| | | | loss1_per_pass(rt)=1-power1/last_power2
| | | | last_power1=power1
| | | | loss_per_pass(transits)=1-power/last_power
| | | | last_power=power
| | | | IF auto_scale$="Y" then
| | | | | FOR i=ROUND(LOG10(loss1_per_pass(rt)))-1 to 0
| | | | | | IF loss1_per_pass(rt)>10^i then
| | | | | | | EXIT IF
| | | | | | ELSE
| | | | | | | loss_max1=10^i
| | | | | | | loss_min1=10^(i-1)
| | | | | | | i=0
| | | | | | END IF
| | | | | NEXT i
| | | | ELSE
| | | | | loss_min1=0.00001
| | | | | loss_max1=1
| | | | END IF
| | | END IF
| | | |
| | | WINDOW #9
| | | SET COLOR 15
| | | loss_min=min(loss_min,loss_min1)
| | | loss_min=min(loss_min,loss_min2)
| | | loss_max=max(loss_max,loss_max1)
| | | loss_max=max(loss_max,loss_max2)
| | | |
| | | IF mirror=2 then
| | | | IF loss_min<>loss_min_old OR loss_max<>loss_max_old then
| | | | | CLEAR
| | | | | WINDOW #1
| | | | | BOX CLEAR 0.339,0.373,0,0.4
```

```
| | | | | CALL labels
| | | | | WINDOW #9
| | | | | SET WINDOW 0,max_transits,LOG10(loss_min),LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7
| | | | | SET COLOR 7
| | | | | FOR i=0 to max_transits STEP max(1,INT(max_transits/10))
| | | | | | PLOT i,LOG10(loss_min);i,LOG10(loss_max)
| | | | | NEXT i
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | FOR i=ROUND(LOG10(loss_min)) to ROUND(LOG10(loss_max))
| | | | | | SET COLOR 7
| | | | | | PLOT 0,i;max_transits,i
| | | | | | SET COLOR 15
| | | | | | PLOT TEXT, AT -max_transits/40,i:STR$(i+2)
| | | | | NEXT i
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | PLOT TEXT, AT max_transits/2,LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7:"P.L./Pass/Mirror"
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | PLOT 0,LOG10(loss_min);0,LOG10(loss_max);
| | | | | PLOT max_transits,LOG10(loss_max);max_transits,LOG10(loss_min);
| | | | | PLOT 0,LOG10(loss_min)
| | | | | IF loss_min<0 and loss_max>0 then PLOT 0,0;max_transits,0
| | | | | SET COLOR 12
| | | | | FOR i=1 to transits
| | | | | | PLOT i,LOG10(loss_per_pass(i));
| | | | | NEXT i
| | | | | PLOT
| | | | | IF transits>1 then
| | | | | | SET COLOR 10                                   ! M1.
| | | | | | FOR i=2 to 2*rt-2 STEP 2
| | | | | | | PLOT i,LOG10(loss1_per_pass(i/2));
| | | | | | NEXT i
| | | | | END IF
| | | | | PLOT
| | | | | SET COLOR 11                                   ! M2.
| | | | | FOR i=1 to 2*rt-1 STEP 2
| | | | | | PLOT i,LOG10(loss2_per_pass((i+1)/2));
| | | | | NEXT i
| | | | | PLOT
| | | | ELSE
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | BOX CLEAR 0,max_transits,LOG10(loss_max)+(LOG10(loss_max)
| | | | | | -LOG10(loss_min))/15,LOG10(loss_max)+(LOG10(loss_max)
```

```
| | | | | | -LOG10(loss_min))/5
| | | | | PLOT TEXT, AT max_transits/2,LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7:"P.L./Pass/Mirror"
| | | | | SET COLOR 12
| | | | | PLOT transits-1,LOG10(loss_per_pass(transits-1));
| | | | | PLOT transits,LOG10(loss_per_pass(transits))
| | | | | SET COLOR 11
| | | | | PLOT 2*rt-3,LOG10(loss2_per_pass(rt-1));
| | | | | PLOT 2*rt-1,LOG10(loss2_per_pass(rt))
| | | | END IF
| | | | WINDOW #15
| | | | SET TEXT JUSTIFY "center","half"
| | | | BOX CLEAR 0.63,0.95,0,1
| | | | SET COLOR 11
| | | | PLOT TEXT, AT 0.776,0.5:"M2              "
| | | | SET COLOR 15
| | | | PLOT TEXT, AT 0.776,0.5:&
| | | | &USING$("    P.L.:+#.##^^^ #",100*loss2_per_pass(rt),"%")
| | | | SET COLOR 15
| | | | PLOT TEXT, AT 0.224,0.5:"M1              "
| | | | BOX CLEAR 0.35,0.63,0,1
| | | | PLOT TEXT, AT 0.488,0.5:&
| | | | & USING$("Final P.L.:+#.##^^^ #",100*loss_per_pass(transits),"%")
| | | | |
| | | ELSEIF mirror=1 then
| | | | IF loss_min<>loss_min_old OR loss_max<>loss_max_old then
| | | | | CLEAR
| | | | | WINDOW #1
| | | | | BOX CLEAR 0.339,0.373,0,0.4
| | | | | CALL labels
| | | | | WINDOW #9
| | | | | SET WINDOW 0,max_transits,LOG10(loss_min),LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7
| | | | | SET COLOR 7
| | | | | FOR i=0 to max_transits STEP max(1,INT(max_transits/10))
| | | | | | PLOT i,LOG10(loss_min);i,LOG10(loss_max)
| | | | | NEXT i
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | FOR i=ROUND(LOG10(loss_min)) to ROUND(LOG10(loss_max))
| | | | | | SET COLOR 7
| | | | | | PLOT 0,i;max_transits,i
| | | | | | SET COLOR 15
| | | | | | PLOT TEXT, AT -max_transits/40,i:STR$(i+2)
| | | | | NEXT i
```

```
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | PLOT TEXT, AT max_transits/2,LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7:"P.L./Pass/Mirror"
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | PLOT 0,LOG10(loss_min);0,LOG10(loss_max);
| | | | | PLOT max_transits,LOG10(loss_max);max_transits,LOG10(loss_min);
| | | | | PLOT 0,LOG10(loss_min)
| | | | | IF loss_min<0 and loss_max>0 then PLOT 0,0;max_transits,0
| | | | | SET COLOR 12
| | | | | FOR i=1 to transits
| | | | | | PLOT i,LOG10(loss_per_pass(i));
| | | | | NEXT i
| | | | | PLOT
| | | | | SET COLOR 10                                    ! M1
| | | | | FOR i=2 to 2*rt STEP 2
| | | | | | PLOT i,LOG10(loss1_per_pass(i/2));
| | | | | NEXT i
| | | | | PLOT
| | | | | SET COLOR 11                                    ! M2
| | | | | FOR i=1 to 2*rt-1 STEP 2
| | | | | | PLOT i,LOG10(loss2_per_pass((i+1)/2));
| | | | | NEXT i
| | | | | PLOT
| | | | ELSE
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | BOX CLEAR 0,max_transits,LOG10(loss_max)+(LOG10(loss_max)
| | | | | | -LOG10(loss_min))/15,LOG10(loss_max)+(LOG10(loss_max)
| | | | | | -LOG10(loss_min))/5
| | | | | PLOT TEXT, AT max_transits/2,LOG10(loss_max)
| | | | | | +(LOG10(loss_max)-LOG10(loss_min))/7:"P.L./Pass/Mirror"
| | | | | SET COLOR 12
| | | | | PLOT transits-1,LOG10(loss_per_pass(transits-1));
| | | | | PLOT transits,LOG10(loss_per_pass(transits))
| | | | | SET COLOR 10
| | | | | IF transits=2 then
| | | | | | PLOT 2*rt,LOG10(loss1_per_pass(rt))
| | | | | ELSE
| | | | | | PLOT 2*rt-2,LOG10(loss1_per_pass(rt-1));
| | | | | | PLOT 2*rt,LOG10(loss1_per_pass(rt))
| | | | | END IF
| | | | END IF
| | | | WINDOW #15
| | | | SET TEXT JUSTIFY "center","half"
| | | | BOX CLEAR 0,0.35,0,1
```

```
| | | | SET COLOR 10
| | | | PLOT TEXT, AT 0.224,0.5:"M1            "
| | | | SET COLOR 15
| | | | PLOT TEXT, AT 0.224,0.5:&
| | | | & USING$("   P.L.:+#.##^^^ #",100*loss1_per_pass(rt),"%")
| | | | SET COLOR 15
| | | | PLOT TEXT, AT 0.776,0.5:"M2            "
| | | | BOX CLEAR 0.35,0.63,0,1
| | | | PLOT TEXT, AT 0.488,0.5:&
| | | | & USING$("Final P.L.:+#.##^^^ #",100*loss_per_pass(transits),"%")
| | | END IF
| | | loss_min_old=loss_min
| | | loss_max_old=loss_max
| | | |
| | | |              ! Plot Mirror2's amplitude variation per Round Trip(RT) below.
| | | |
| | | IF transits=3 then                                    ! Mirror=2.
| | | | intvar2_per_pass(1),intvar2_max,intvar2_min=int2_amp/last_int2_amp-1
| | | | |
| | | ELSEIF mirror=2 and transits>3 then                   ! Mirror2 info only.
| | | | WINDOW #10
| | | | SET COLOR 15
| | | | intvar2_per_pass(rt-1)=int2_amp/last_int2_amp-1
| | | | intvar2_min=min(intvar2_min,intvar2_per_pass(rt-1))
| | | | intvar_min= min(intvar_min,intvar1_min)
| | | | intvar_min= min(intvar_min,intvar2_min)
| | | | intvar2_max=max(intvar2_max,intvar2_per_pass(rt-1))
| | | | intvar_max= max(intvar_max,intvar1_max)
| | | | intvar_max= max(intvar_max,intvar2_max)
| | | | IF intvar_min<>intvar2_min_old OR intvar_max<>intvar2_max_old then
| | | | | CLEAR
| | | | | WINDOW #1
| | | | | BOX CLEAR 0.603,0.661,0,0.4
| | | | | CALL labels
| | | | | WINDOW #10
| | | | | SET WINDOW 0,max_rt,intvar_min-(intvar_max-intvar_min)/25,
| | | | | | intvar_max+(intvar_max-intvar_min)/7
| | | | | SET COLOR 7
| | | | | FOR i=0 to max_rt STEP max(1,INT(max_rt/10))
| | | | | | PLOT i,intvar_min;i,intvar_max
| | | | | NEXT i
| | | | | SET COLOR 15
| | | | | BOX CLEAR 0,max_rt,intvar_max+(intvar_max-intvar_min)/15,
| | | | | | intvar_max+(intvar_max-intvar_min)/5
```

```
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/7:&
| | | | | & USING$("Decay/RT:+#.#^^^",100*intvar2_per_pass(rt-1))
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | i=-max_rt/40
| | | | | PLOT TEXT, AT i,intvar_max:STR$(ROUND(100*intvar_max))
| | | | | PLOT TEXT, AT i,intvar_min:STR$(ROUND(100*intvar_min))
| | | | | PLOT TEXT, AT i,intvar_min+(intvar_max-intvar_min)/2:"% "
| | | | | PLOT 0,intvar_min;0,intvar_max;
| | | | | PLOT max_rt,intvar_max;max_rt,intvar_min;0,intvar_min
| | | | | IF intvar_min<0 and intvar_max>0 then
| | | | | | SET COLOR 7
| | | | | | PLOT 0,0;max_rt,0
| | | | | END IF
| | | | | SET COLOR 11
| | | | | FOR i=1 to rt-1
| | | | | | PLOT i,intvar2_per_pass(i);
| | | | | NEXT i
| | | | | PLOT
| | | | ELSE
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | BOX CLEAR 0.57*max_rt,max_rt,intvar_max+(intvar_max-
| | | | | | intvar_min)/15,intvar_max+(intvar_max-intvar_min)/5
| | | | | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/7:&
| | | | | & USING$("Decay/RT:+#.#^^^",100*intvar2_per_pass(rt-1))
| | | | | SET COLOR 11
| | | | | PLOT rt-2,intvar2_per_pass(rt-2);
| | | | | PLOT rt-1,intvar2_per_pass(rt-1)
| | | | END IF
| | | | last_int2_amp=int2_amp
| | | | intvar_max_old,intvar2_max_old=intvar_max
| | | | intvar_min_old,intvar2_min_old=intvar_min
| | | | |
| | | | |            ! Plot Mirror1's amplitude variation per Round Trip(RT) below.
| | | | |
| | | ELSEIF transits=2 then                                  ! Mirror=1.
| | | | intvar1_per_pass(1),intvar1_max,intvar1_min=int1_amp/last_int1_amp-1
| | | | intvar1_min_old,intvar2_min_old=1e10
| | | | intvar1_max_old,intvar2_max_old=-1e10
| | | | |
| | | ELSEIF mirror=1 then                                    ! Mirror1 info only.
| | | | WINDOW #8
| | | | SET COLOR 15
| | | | intvar1_per_pass(rt)=int1_amp/last_int1_amp-1
```

```
| | | | intvar1_min=min(intvar1_min,intvar1_per_pass(rt))
| | | | intvar_min= min(intvar_min,intvar1_min)
| | | | intvar_min= min(intvar_min,intvar2_min)
| | | | intvar1_max=max(intvar1_max,intvar1_per_pass(rt))
| | | | intvar_max= max(intvar_max,intvar1_max)
| | | | intvar_max= max(intvar_max,intvar2_max)
| | | | IF intvar_min<>intvar1_min_old OR intvar_max<>intvar1_max_old then
| | | | | CLEAR
| | | | | WINDOW #1
| | | | | BOX CLEAR 0.05,0.109,0,0.4
| | | | | CALL labels
| | | | | WINDOW #8
| | | | | SET WINDOW 0,max_rt,intvar_min-(intvar_max-intvar_min)/25,
| | | | | | intvar_max+(intvar_max-intvar_min)/7
| | | | | SET COLOR 7
| | | | | FOR i=0 to max_rt STEP max(1,INT(max_rt/10))
| | | | | | PLOT i,intvar_min;i,intvar_max
| | | | | NEXT i
| | | | | SET COLOR 15
| | | | | BOX CLEAR 0,max_rt,intvar_max+(intvar_max-intvar_min)/15,
| | | | | | intvar_max+(intvar_max-intvar_min)/5
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/7:&
| | | | | & USING$("Decay/RT:+#.#^^^",100*intvar1_per_pass(rt))
| | | | | SET TEXT JUSTIFY "right","half"
| | | | | i=-max_rt/40
| | | | | PLOT TEXT, AT i,intvar_max:STR$(ROUND(100*intvar_max))
| | | | | PLOT TEXT, AT i,intvar_min:STR$(ROUND(100*intvar_min))
| | | | | PLOT TEXT, AT i,intvar_min+(intvar_max-intvar_min)/2:"% "
| | | | | PLOT 0,intvar_min;0,intvar_max;
| | | | | PLOT max_rt,intvar_max;max_rt,intvar_min;0,intvar_min
| | | | | IF intvar_min<0 and intvar_max>0 then
| | | | | | SET COLOR 7
| | | | | | PLOT 0,0;max_rt,0
| | | | | END IF
| | | | | SET COLOR 10
| | | | | FOR i=1 to rt
| | | | | | PLOT i,intvar1_per_pass(i);
| | | | | NEXT i
| | | | | PLOT
| | | | ELSE
| | | | | SET TEXT JUSTIFY "center","top"
| | | | | BOX CLEAR 0.57*max_rt,max_rt,intvar_max+(intvar_max
| | | | | | -intvar_min) /15,intvar_max+(intvar_max-intvar_min)/5
```

```
| | | | | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/7:&
| | | | | & USING$("Decay/RT:+#.#^^^",100*intvar1_per_pass(rt))
| | | | | SET COLOR 10
| | | | | PLOT rt-1,intvar1_per_pass(rt-1);
| | | | | PLOT rt,intvar1_per_pass(rt)
| | | | END IF
| | | | last_int1_amp=int1_amp
| | | | intvar_max_old,intvar1_max_old=intvar_max
| | | | intvar_min_old,intvar1_min_old=intvar_min
| | | END IF
| | | |
| | NEXT transits                                         ! End of main loop.
| | |
| | |                                  ! Halt the program for a full screen view.
| | |
| | transits=transits-1
| | WINDOW #14
| | SET WINDOW 0,incr+3,0,2
| | SET TEXT JUSTIFY "center","half"
| | CLEAR
| | GET KEY key
| | FOR i=8 to 15
| | | CLOSE # i
| | NEXT i
| | |
| | |                                ! Halt the program for a smaller screen view
| | |                                ! for use in the windows clipboard.
| | |
| | OPEN #8 : SCREEN 0.110,0.338,0.19,0.4                  !!!
| | OPEN #9 : SCREEN 0.374,0.602,0.19,0.4                    !
| | OPEN #10: SCREEN 0.662,0.890,0.19,0.4                    !
| | OPEN #11: SCREEN 0.110,0.338,0.15,0.18                   ! Screen
| | OPEN #12: SCREEN 0.374,0.602,0.15,0.18                   ! viewports.
| | OPEN #13: SCREEN 0.662,0.890,0.15,0.18                   !
| | OPEN #14: SCREEN 0.090,0.935,0.815,0.835                 !
| | OPEN #15: SCREEN 0.05,0.95,0.11,0.15                   !!!
| | WINDOW #15
| | SET WINDOW 0.05,0.95,0,1
| | WINDOW #1
| | BOX CLEAR 0,1,0,0.4
| | CALL labels
| | |
| | |                                ! Plot Power Loss (P.L.) per pass below.
| | |
```

```
| | WINDOW #9
| | SET COLOR 7
| | SET WINDOW 0,max_transits,LOG10(loss_min)-(LOG10(loss_max)
| | | | -LOG10(loss_min))/25,LOG10(loss_max)+(LOG10(loss_max)
| | | | -LOG10(loss_min))/5
| | FOR i=0 to max_transits STEP max(1,INT(max_transits/10))
| | | PLOT i,LOG10(loss_min);i,LOG10(loss_max)
| | NEXT i
| | SET TEXT JUSTIFY "right","half"
| | FOR i=ROUND(LOG10(loss_min)) to ROUND(LOG10(loss_max))
| | | SET COLOR 7
| | | PLOT 0,i;max_transits,i
| | | SET COLOR 15
| | | PLOT TEXT, AT -max_transits/40,i:STR$(i+2)
| | NEXT i
| | SET TEXT JUSTIFY "center","top"
| | PLOT TEXT, AT max_transits/2,LOG10(loss_max)+(LOG10(loss_max)
| | | -LOG10(loss_min))/5:"P.L./Pass/Mirror"
| | SET TEXT JUSTIFY "right","half"
| | PLOT 0,LOG10(loss_min);0,LOG10(loss_max);
| | PLOT max_transits,LOG10(loss_max);max_transits,LOG10(loss_min);
| | PLOT 0,LOG10(loss_min)
| | IF loss_min<0 and loss_max>0 then
| | | SET COLOR 7
| | | PLOT 0,0;max_transits,0
| | END IF
| | SET COLOR 12
| | FOR i=1 to transits
| | | PLOT i,LOG10(loss_per_pass(i));
| | NEXT i
| | PLOT
| | SET COLOR 10                                          ! M1
| | FOR i=2 to 2*rt STEP 2
| | | PLOT i,LOG10(loss1_per_pass(i/2));
| | NEXT i
| | PLOT
| | SET COLOR 11                                          ! M2
| | FOR i=1 to 2*rt-1 STEP 2
| | | PLOT i,LOG10(loss2_per_pass((i+1)/2));
| | NEXT i
| | PLOT
| | WINDOW #15
| | SET TEXT JUSTIFY "center","half"
| | SET COLOR 15
```

```
| | PLOT TEXT, AT 0.224,0.5:&
| | & USING$("M1 P.L.:+#.##^^^ #",100*loss1_per_pass(rt),"%")
| | SET COLOR 15
| | PLOT TEXT, AT 0.776,0.5:&
| | & USING$("M2 P.L.:+#.##^^^ #",100*loss2_per_pass(rt-1),"%")
| | PLOT TEXT, AT 0.488,0.5:&
| | & USING$("Final P.L.:+#.##^^^ #",100*loss_per_pass(transits),"%")
| | |
| | |               ! Plot Mirror1's amplitude variation per Round Trip(RT) below.
| | |
| | WINDOW #8
| | SET COLOR 7
| | SET WINDOW 0,max_rt,intvar_min-(intvar_max-intvar_min)/25,
| | | intvar_max+(intvar_max-intvar_min)/5
| | FOR i=0 to max_rt STEP max(1,INT(max_rt/10))
| | | PLOT i,intvar_min;i,intvar_max
| | NEXT i
| | SET COLOR 15
| | SET TEXT JUSTIFY "center","top"
| | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/5:&
| | & USING$("Decay/RT:+#.#^^^^",100*intvar1_per_pass(rt))
| | SET TEXT JUSTIFY "right","half"
| | i=-max_rt/40
| | PLOT TEXT, AT i,intvar_max:STR$(ROUND(100*intvar_max))
| | PLOT TEXT, AT i,intvar_min:STR$(ROUND(100*intvar_min))
| | PLOT TEXT, AT i,intvar_min+(intvar_max-intvar_min)/2:"% "
| | PLOT 0,intvar_min;0,intvar_max;
| | PLOT max_rt,intvar_max;max_rt,intvar_min;0,intvar_min
| | IF intvar_min<0 and intvar_max>0 then
| | | SET COLOR 7
| | | PLOT 0,0;max_rt,0
| | END IF
| | SET COLOR 10
| | FOR i=1 to rt
| | | PLOT i,intvar1_per_pass(i);
| | NEXT i
| | PLOT
| | |
| | |               ! Plot Mirror2's amplitude variation per Round Trip (RT) below.
| | |
| | WINDOW #10
| | SET COLOR 7
| | SET WINDOW 0,max_rt,intvar_min-(intvar_max-intvar_min)/25,
| | | intvar_max+(intvar_max-intvar_min)/5
```

```
| | FOR i=0 to max_rt STEP max(1,INT(max_rt/10))
| | | PLOT i,intvar_min;i,intvar_max
| | NEXT i
| | SET COLOR 15
| | SET TEXT JUSTIFY "center","top"
| | PLOT TEXT, AT max_rt/2,intvar_max+(intvar_max-intvar_min)/5:&
| | & USING$("Decay/RT:+#.#^^^",100*intvar2_per_pass(rt-1))
| | SET TEXT JUSTIFY "right","half"
| | i=-max_rt/40
| | PLOT TEXT, AT i,intvar_max:STR$(ROUND(100*intvar_max))
| | PLOT TEXT, AT i,intvar_min:STR$(ROUND(100*intvar_min))
| | PLOT TEXT, AT i,intvar_min+(intvar_max-intvar_min)/2:"% "
| | PLOT 0,intvar_min;0,intvar_max;
| | PLOT max_rt,intvar_max;max_rt,intvar_min;0,intvar_min
| | IF intvar_min<0 and intvar_max>0 then
| | | SET COLOR 7
| | | PLOT 0,0;max_rt,0
| | END IF
| | SET COLOR 11
| | FOR i=1 to rt-1
| | | PLOT i,intvar2_per_pass(i);
| | NEXT i
| | PLOT
| | CLOSE #14
| | OPEN #14: SCREEN 0.04,0.95,0.815,0.835
| | WINDOW #14
| | SET COLOR 15
| | SET WINDOW 0.04,0.95,0,2
| | SET TEXT JUSTIFY "left","half"
| | diameter$=USING$("M1 Radius=#.##^^^ mm",max_radius1/1e-3)
| | PLOT TEXT, AT 0.0478,1:diameter$
| | diameter$=USING$("M2 Radius=#.##^^^ mm",max_radius2/1e-3)
| | PLOT TEXT, AT 0.6732,1:diameter$
| | SET TEXT JUSTIFY "center","half"
| | PLOT TEXT, AT 0.4856,1:resonator$
| | GET KEY key
| LOOP
| |
END SUB
|
!*********************************************************** !
!                SUBROUTINE LABELS: Labels the x-axis.        !
!*********************************************************** !
|
```

```
SUB labels
| WINDOW #11
| SET WINDOW 0,max_rt,0,1
| SET TEXT JUSTIFY "center","bottom"
| SET COLOR 15
| PLOT TEXT, AT 0,0:"0"
| PLOT TEXT, AT max_rt/2,0:STR$(max_rt/2)
| PLOT TEXT, AT max_rt,0:STR$(max_rt)
| WINDOW #12
| SET WINDOW 0,max_transits,0,1
| SET TEXT JUSTIFY "center","bottom"
| SET COLOR 15
| PLOT TEXT, AT 0,0:"0"
| PLOT TEXT, AT max_transits/2,0:STR$(max_transits/2)
| PLOT TEXT, AT max_transits,0:STR$(max_transits)
| WINDOW #13
| SET WINDOW 0,max_rt,0,1
| SET TEXT JUSTIFY "center","bottom"
| SET COLOR 15
| PLOT TEXT, AT 0,0:"0"
| PLOT TEXT, AT max_rt/2,0:STR$(max_rt/2)
| PLOT TEXT, AT max_rt,0:STR$(max_rt)
END SUB
|
!*********************************************************** !
!       SUBROUTINE VARIABLE_CHANGE: Change some variables    !
!*********************************************************** !
|
SUB variable_change
| SET MODE "80"
| CLEAR
| SET CURSOR 1,1
| PRINT "************************ RESONATE VERSION 1.0
| | **********************"
| PRINT
| PRINT "       COMMENT          INPUT    DEFAULT  VARIABLE
| | NAME"
| PRINT "M1 Type [S,PAR,PL,F,T]- - - -:";TAB(41);m1$;TAB(51);
| | "m_1$={S,PAR,PL,F,T}"
| PRINT "M1 [S,PAR,F,T] R. of Curv.(m):";
| | TAB(40);USING$("+##.####",radius1);TAB(51);"radius1={#}"
| PRINT "M1 [F,T] # of Fresnel Zones -:";TAB(40);zones1;
| | TAB(51);"zones1={#}"
| PRINT "M1 [F,T] # of Tiers per Zone :";TAB(40);tpz1;TAB(51);"tpz1={#}"
```

```
| PRINT "M1 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers1;TAB(51);
| | "outer_tiers1={#}"
| PRINT "M1 [S,PAR,PL] Radial Dim.(mm):";TAB(40);
| | USING$("##.####",max_radius1/1e-3);TAB(51);"max_radius1={#}"
| PRINT
| PRINT "M2 Type {S,PAR,PL,F,T}- - - -:";TAB(41);m2$;TAB(51);
| | "m_2$={S,PAR,PL,F,T}"
| PRINT "M2 [S,PAR,F,T] R. of Curv.(m):";TAB(40);
| | USING$("+##.####",radius2);TAB(51);"radius2={#}"
| PRINT "M2 [F,T] # of Fresnel Zones -:";TAB(40);zones2;
| | TAB(51);"zones2={#}"
| PRINT "M2 [F,T] # of Tiers per Zone :";TAB(40);tpz2;TAB(51);"tpz2={#}"
| PRINT "M2 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers2;TAB(51);
| | "outer_tiers2={#}"
| PRINT "M2 [S,PAR,PL] Radial Dim.(mm):";TAB(40);USING$("##.####",
| | max_radius2/1e-3);TAB(51);"max_radius2={#}"
| PRINT
| PRINT "Wavelength (um) - - - - - - -:";TAB(40);USING$("##.####",
| | lambda/1e-6);TAB(51);"lambda={#}"
| PRINT "Cavity Length (m) - - - - - -:";TAB(40);USING$("##.####",
| | cavity_length);TAB(51);"cavity_length={#}"
| PRINT "# of Round Trips, ie RT - - -:";TAB(40);max_rt;TAB(51);
| | "max_rt={#}"
| PRINT "# of Radial Increments  - - -:";TAB(40);incr;TAB(51);"incr={#}"
| PRINT "# of Azimuthal increments - -:";TAB(40);inct;TAB(51);"inct={#}"
| PRINT "Input Wave {PL,GAUSS} - - - -:";TAB(41);input_wave$;TAB(51);
| | "input_wave={PL,GAUSS}"
| rrow=4
| DO
| | SELECT CASE rrow
| | CASE IS <=4
| | | rrow=4
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M1 Type [S,PAR,PL,F,T]- - - -:";TAB(41);m1$;TAB(51);
| | | | "m_1$={S,PAR,PL,F,T}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M1 Type [S,PAR,PL,F,T]- - - -:";TAB(41);m1$
| | | SET COLOR "white/black"
| | | SELECT CASE m1$
| | | CASE "SPHERICAL"
| | | | m_1$="S"
```

```
2175  | | | CASE "PARABOLIC"
2176  | | | | m_1$="PAR"
2177  | | | CASE "PLANE"
2178  | | | | m_1$="PL"
2179  | | | CASE "TIERED"
2180  | | | | m_1$="T"
2181  | | | CASE "FRESNEL"
2182  | | | | m_1$="F"
2183  | | | CASE ELSE
2184  | | | END SELECT
2185  | | | m_1$=UCASE$(input_string$(rrow,32,"S","PAR","PL","F","T",
      | | | | "Enter either S,PAR,PL,F, or T please",m_1$))
2186  | | | SELECT CASE m_1$
2187  | | | CASE "S"
2188  | | | | m1$="SPHERICAL"
2189  | | | CASE "PAR"
2190  | | | | m1$="PARABOLIC"
2191  | | | CASE "PL"
2192  | | | | m1$="PLANE"
2193  | | | CASE "T"
2194  | | | | m1$="TIERED"
2195  | | | CASE "F"
2196  | | | | m1$="FRESNEL"
2197  | | | CASE ELSE
2198  | | | END SELECT
2199  | | | SET CURSOR 4,1
2200  | | | PRINT erase_line$
2201  | | | SET CURSOR 4,1
2202  | | | PRINT "M1 Type [S,PAR,PL,F,T]- - - -:";TAB(32);m_1$;
      | | | | TAB(41);m1$;TAB(51);"m_1$={S,PAR,PL,F,T}"
2203  | | CASE 5
2204  | | | SET CURSOR rrow,1
2205  | | | PRINT erase_line$
2206  | | | SET CURSOR rrow,1
2207  | | | PRINT "M1 [S,PAR,F,T] R. of Curv.(m):";
      | | | | TAB(40);USING$("+##.####",radius1);TAB(51);"radius1={#}"
2208  | | | SET CURSOR rrow,1
2209  | | | SET COLOR "black/white"
2210  | | | PRINT "M1 [S,PAR,F,T] R. of Curv.(m):";
      | | | | TAB(40);USING$("+##.####",radius1)
2211  | | | SET COLOR "white/black"
2212  | | | temp=input_num(rrow,32,radius1)
2213  | | | IF temp=0 then
2214  | | | | CALL zero
```

```
| | | | rrow=5
| | | ELSEIF temp<0 and (m1$="FRESNEL" OR m1$="TIERED") then
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be POSITIVE when m1$ is either
| | | | | FRESNEL or TIERED, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
| | | | rrow=5
| | | ELSE
| | | | radius1=temp
| | | END IF
| | | SET CURSOR 5,1
| | | PRINT erase_line$
| | | SET CURSOR 5,1
| | | PRINT "M1 [S,PAR,F,T] R. of Curv.(m):";TAB(31);USING$("+##.####",
| | | | radius1);TAB(40);USING$("+##.####",radius1);TAB(51);"radius1={#}"
| | CASE 6
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M1 [F,T] # of Zones - - - - -:";TAB(40);zones1;
| | | | TAB(51);"zones1={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M1 [F,T] # of Zones - - - - -:";TAB(40);zones1
| | | SET COLOR "white/black"
| | | zones1=INT(ABS(input_num(rrow,32,zones1)))
| | | SET CURSOR 6,1
| | | PRINT erase_line$
| | | SET CURSOR 6,1
| | | PRINT "M1 [F,T] # of Zones - - - - -:";TAB(31);zones1;TAB(40);zones1;
| | | | TAB(51);"zones1={#}"
| | CASE 7
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M1 [F,T] # of Tiers per Zone :";TAB(40);tpz1;
| | | | TAB(51);"tpz1={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M1 [F,T] # of Tiers per Zone :";TAB(40);tpz1
```

```
| | | SET COLOR "white/black"
| | | tpz1=INT(ABS(input_num(rrow,32,tpz1)))
| | | IF MOD(tpz1,2)=0 then                          ! Odd # of tiers only.
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be an ODD number, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
| | | | rrow=7
| | | END IF
| | | SET CURSOR 7,1
| | | PRINT erase_line$
| | | SET CURSOR 7,1
| | | PRINT "M1 [F,T] # of Tiers per Zone :";TAB(31);tpz1;TAB(40);tpz1;
| | | | TAB(51);"tpz1={#}"
| | CASE 8
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M1 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers1;
| | | | TAB(51);"outer_tiers1={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M1 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers1
| | | SET COLOR "white/black"
| | | outer_tiers1=INT(ABS(input_num(rrow,32,outer_tiers1)))
| | | SET CURSOR 8,1
| | | PRINT erase_line$
| | | SET CURSOR 8,1
| | | PRINT "M1 [F,T] # of outer Tiers - -:";TAB(31);outer_tiers1;
| | | | TAB(40);outer_tiers1;TAB(51);"outer_tiers1={#}"
| | CASE 9
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M1 [S,PAR,PL] Radial Dim.(mm):";TAB(40);
| | | | USING$("##.####",max_radius1/1e-3);TAB(51);"max_radius1={#}"
| | | SET CURSOR rrow,1
| | | IF MOD(tpz1,2)=0 then tpz1=tpz1+1              ! Odd # of tiers only.
| | | IF zones1=0 then
| | | | tiers1=outer_tiers1
| | | ELSE
```

```
| | | | tiers1=tpz1*(zones1-0.5)+0.5+outer_tiers1                    ! Total tiers.
| | | END IF
| | | IF m_1$="F" OR m_1$="T" then
| | | | IF tiers1>0 then
| | | | | max_radius1=SQR(ABS(radius1)*(2*tiers1-2)*lambda/2/tpz1
| | | | | | +((2*tiers1-2)*lambda/2/tpz1)^2)
| | | | | SET COLOR "black/white"
| | | | | PRINT "M1 [S,PAR,PL] Radial Dim.(mm):";
| | | | | | TAB(40);USING$("##.####",max_radius1/1e-3)
| | | | | SET COLOR "white/black"
| | | | | maxradius1=1e-3*ABS(input_num(rrow,32,max_radius1/1e-3))
| | | | ELSE
| | | | | SET COLOR "black/white"
| | | | | SET CURSOR 2,1
| | | | | PRINT "Both zones1 & outer_tiers1 MUST NOT equal 0,
| | | | | |  Cr to continue:";
| | | | | SET COLOR "white/black"
| | | | | CALL input_
| | | | | SET CURSOR 2,1
| | | | | PRINT erase_line$
| | | | | rrow=6
| | | | END IF
| | | ELSE
| | | | IF tiers1>0 then max_radius1=SQR(ABS(radius1)*(2*tiers1-2)
| | | | | *lambda/2/tpz1+((2*tiers1-2)*lambda/2/tpz1)^2)
| | | | SET COLOR "black/white"
| | | | PRINT "M1 [S,PAR,PL] Radial Dim.(mm):";
| | | | | TAB(40);USING$("##.####",max_radius1/1e-3)
| | | | SET COLOR "white/black"
| | | | temp=1e-3*ABS(input_num(rrow,32,max_radius1/1e-3))
| | | | IF temp=0 then
| | | | | CALL zero
| | | | | rrow=9
| | | | ELSE
| | | | | max_radius1=temp
| | | | END IF
| | | END IF
| | | SET CURSOR 9,1
| | | PRINT erase_line$
| | | SET CURSOR 9,1
| | | PRINT "M1 [S,PAR,PL] Radial Dim.(mm):";TAB(31);
| | | | USING$("##.####",max_radius1/1e-3);TAB(40);
| | | | USING$("##.####",max_radius1/1e-3);TAB(51);"max_radius1={#}"
| | CASE 11
```

```
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 Type [S,PAR,PL,F,T]- - - -:";TAB(41);m2$;TAB(51);
| | | | "m_2$={S,PAR,PL,F,T}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M2 Type [S,PAR,PL,F,T]- - - -:";TAB(41);m2$
| | | SET COLOR "white/black"
| | | SELECT CASE m2$
| | | CASE "SPHERICAL"
| | | | m_2$="S"
| | | CASE "PARABOLIC"
| | | | m_2$="PAR"
| | | CASE "PLANE"
| | | | m_2$="PL"
| | | CASE "TIERED"
| | | | m_2$="T"
| | | CASE "FRESNEL"
| | | | m_2$="F"
| | | CASE ELSE
| | | END SELECT
| | | m_2$=input_string$(rrow,32,"S","PAR","PL","F","T",
| | | | "Enter either S,PAR,PL,F, or T please",m_2$)
| | | SELECT CASE m_2$
| | | CASE "S"
| | | | m2$="SPHERICAL"
| | | CASE "PAR"
| | | | m2$="PARABOLIC"
| | | CASE "PL"
| | | | m2$="PLANE"
| | | CASE "T"
| | | | m2$="TIERED"
| | | CASE "F"
| | | | m2$="FRESNEL"
| | | CASE ELSE
| | | END SELECT
| | | SET CURSOR 11,1
| | | PRINT erase_line$
| | | SET CURSOR 11,1
| | | PRINT "M2 Type [S,PAR,PL,F,T]- - - -:";TAB(32);m_2$;
| | | | TAB(41);m2$;TAB(51);"m_2$={S,PAR,PL,F,T}"
| | CASE 12
| | | SET CURSOR rrow,1
```

```
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 [S,PAR,F,T] R. of Curv.(m):";
| | | | TAB(40);USING$("+##.####",radius2);TAB(51);"radius2={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M2 [S,PAR,F,T] R. of Curv.(m):";
| | | | TAB(40);USING$("+##.####",radius2)
| | | SET COLOR "white/black"
| | | temp=input_num(rrow,32,radius2)
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=12
| | | ELSEIF temp<0 and (m2$="FRESNEL" OR m2$="TIERED") then
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be POSITIVE when m2$ is either
| | | | | FRESNEL or TIERED, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
| | | | rrow=12
| | | ELSE
| | | | radius2=temp
| | | END IF
| | | SET CURSOR 12,1
| | | PRINT erase_line$
| | | SET CURSOR 12,1
| | | PRINT "M2 [S,PAR,F,T] R. of Curv.(m):";TAB(31);USING$("+##.####",
| | | | radius2);TAB(40);USING$("+##.####",radius2);TAB(51);"radius2={#}"
| | CASE 13
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 [F,T] # of Zones - - - - -:";TAB(40);zones2;TAB(51);
| | | | "zones2={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M2 [F,T] # of Zones - - - - -:";TAB(40);zones2
| | | SET COLOR "white/black"
| | | zones2=INT(ABS(input_num(rrow,32,zones2)))
| | | SET CURSOR 13,1
| | | PRINT erase_line$
```

```
| | | SET CURSOR 13,1
| | | PRINT "M2 [F,T] # of Zones - - - - -:";TAB(31);zones2;TAB(40);zones2;
| | | | TAB(51);"zones2={#}"
| | CASE 14
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 [F,T] # of Tiers per Zone :";TAB(40);tpz2;
| | | | TAB(51);"tpz2={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M2 [F,T] # of Tiers per Zone :";TAB(40);tpz2
| | | SET COLOR "white/black"
| | | tpz2=INT(ABS(input_num(rrow,32,tpz2)))
| | | IF MOD(tpz2,2)=0 then                                   ! Odd # of tiers only.
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be an ODD number, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
| | | | rrow=14
| | | END IF
| | | SET CURSOR 14,1
| | | PRINT erase_line$
| | | SET CURSOR 14,1
| | | PRINT "M2 [F,T] # of Tiers per Zone :";TAB(31);tpz2;TAB(40);tpz2;
| | | | TAB(51);"tpz2={#}"
| | CASE 15
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers2;TAB(51);
| | | | "outer_tiers2={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "M2 [F,T] # of outer Tiers - -:";TAB(40);outer_tiers2
| | | SET COLOR "white/black"
| | | outer_tiers2=INT(ABS(input_num(rrow,32,outer_tiers2)))
| | | SET CURSOR 15,1
| | | PRINT erase_line$
| | | SET CURSOR 15,1
| | | PRINT "M2 [F,T] # of outer Tiers - -:";TAB(31);outer_tiers2;TAB(40);
```

```
| | | | outer_tiers2;TAB(51);"outer_tiers2={#}"
| | CASE 16
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "M2 [S,PAR,PL] Radial Dim.(mm):";TAB(40);
| | | | USING$("##.####",max_radius2/1e-3);TAB(51);"max_radius2={#}"
| | | SET CURSOR rrow,1
| | | IF MOD(tpz2,2)=0 then tpz2=tpz2+1                    ! Odd # of tiers only.
| | | IF zones2=0 then
| | | | tiers2=outer_tiers2
| | | ELSE
| | | | tiers2=tpz2*(zones2-0.5)+0.5+outer_tiers2                ! Total tiers.
| | | END IF
| | | IF m_2$="F" OR m_2$="T" then
| | | | IF tiers2>0 then
| | | | | max_radius2=SQR(ABS(radius2)*(2*tiers2-2)*lambda/2/tpz2
| | | | | | +((2*tiers2-2)*lambda/2/tpz2)^2)
| | | | | SET COLOR "black/white"
| | | | | PRINT "M2 [S,PAR,PL] Radial Dim.(mm):";TAB(40);
| | | | | | USING$("##.####",max_radius2/1e-3)
| | | | | SET COLOR "white/black"
| | | | | maxradius2=1e-3*ABS(input_num(rrow,32,max_radius2/1e-3))
| | | | ELSE
| | | | | SET COLOR "black/white"
| | | | | SET CURSOR 2,1
| | | | | PRINT "Both zones1 & outer_tiers1 MUST NOT equal 0,
| | | | | | Cr to continue:";
| | | | | SET COLOR "white/black"
| | | | | CALL input_
| | | | | SET CURSOR 2,1
| | | | | PRINT erase_line$
| | | | | rrow=13
| | | | END IF
| | | ELSE
| | | | IF tiers2>0 then  max_radius2=SQR(ABS(radius2)*(2*tiers2-2)*lambda
| | | | | /2/tpz2+((2*tiers2-2)*lambda/2/tpz2)^2)
| | | | SET COLOR "black/white"
| | | | PRINT "M2 [S,PAR,PL] Radial Dim.(mm):";
| | | | | TAB(40);USING$("##.####",max_radius2/1e-3)
| | | | SET COLOR "white/black"
| | | | temp=1e-3*ABS(input_num(rrow,32,max_radius2/1e-3))
| | | | IF temp=0 then
| | | | | CALL zero
```

```
| | | | | rrow=16
| | | | ELSE
| | | | | max_radius2=temp
| | | | END IF
| | | END IF
| | | SET CURSOR 16,1
| | | PRINT erase_line$
| | | SET CURSOR 16,1
| | | PRINT "M2 [S,PAR,PL] Radial Dim.(mm):";TAB(31);USING$("##.####",
| | | | max_radius2/1e-3);TAB(40);USING$("##.####",max_radius2/1e-3);
| | | | TAB(51);"max_radius2={#}"
| | CASE 18
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "Wavelength (um) - - - - - - -:";TAB(40);USING$("##.####",
| | | | lambda/1e-6);TAB(51);"lambda={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "Wavelength (um) - - - - - - -:";TAB(40);USING$("##.####",
| | | | lambda/1e-6)
| | | SET COLOR "white/black"
| | | temp=1e-6*ABS(input_num(rrow,32,lambda/1e-6))
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=18
| | | ELSE
| | | | lambda=temp
| | | END IF
| | | SET CURSOR 18,1
| | | PRINT erase_line$
| | | SET CURSOR 18,1
| | | PRINT "Wavelength (um) - - - - - - -:";TAB(31);USING$("##.####",
| | | | lambda/1e-6);TAB(40);USING$("##.####",lambda/1e-6);
| | | | TAB(51);"lambda={#}"
| | CASE 19
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "Cavity Length (m) - - - - - -:";TAB(40);USING$("##.####",
| | | | cavity_length);TAB(51);"cavity_length={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "Cavity Length (m) - - - - - -:";TAB(40);USING$("##.####",
```

```
| | | | cavity_length)
| | | SET COLOR "white/black"
| | | temp=ABS(input_num(rrow,32,cavity_length))
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=19
| | | ELSE
| | | | cavity_length=temp
| | | END IF
| | | SET CURSOR 19,1
| | | PRINT erase_line$
| | | SET CURSOR 19,1
| | | PRINT "Cavity Length (m) - - - - - -:";TAB(31);USING$("##.####",
| | | | cavity_length);TAB(40);USING$("##.####",cavity_length);
| | | | TAB(51);"cavity_length={#}"
| | CASE 20
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "# of Round Trips, ie RT - - -:";TAB(40);max_rt;TAB(51);
| | | | "max_rt={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "# of Round Trips, ie RT - - -:";TAB(40);max_rt
| | | SET COLOR "white/black"
| | | temp=INT(ABS(input_num(rrow,32,max_rt)))
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=20
| | | ELSE
| | | | max_rt=temp
| | | END IF
| | | SET CURSOR 20,1
| | | PRINT erase_line$
| | | SET CURSOR 20,1
| | | PRINT "# of Round Trips, ie RT - - -:";TAB(31);max_rt;TAB(40);max_rt;
| | | | TAB(51);"max_rt={#}"
| | CASE 21
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "# of Radial Increments  - - -:";TAB(40);incr;TAB(51);"incr={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
```

```
| | | PRINT "# of Radial Increments - - -:";TAB(40);incr
| | | SET COLOR "white/black"
| | | temp=INT(ABS(input_num(rrow,32,incr)))
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=21
| | | ELSEIF MOD(temp,2)=1 then
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be an EVEN number, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
| | | | rrow=21
| | | ELSE
| | | | incr=temp
| | | END IF
| | | SET CURSOR 21,1
| | | PRINT erase_line$
| | | SET CURSOR 21,1
| | | PRINT "# of Radial Increments - - -:";TAB(31);incr;TAB(40);incr;
| | | | TAB(51);"incr={#}"
| | CASE 22
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "# of Azimuthal increments - -:";TAB(40);inct;TAB(51);"inct={#}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "# of Azimuthal increments - -:";TAB(40);inct
| | | SET COLOR "white/black"
| | | temp=INT(ABS(input_num(rrow,32,inct)))
| | | IF temp=0 then
| | | | CALL zero
| | | | rrow=22
| | | ELSEIF MOD(temp,2)=1 then
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "Must be an EVEN number, Cr to continue:";
| | | | CALL input_
| | | | SET CURSOR 2,1
| | | | SET COLOR "white/black"
| | | | PRINT erase_line$
```

```
| | | | rrow=22
| | | ELSE
| | | | inct=temp
| | | END IF
| | | SET CURSOR 22,1
| | | PRINT erase_line$
| | | SET CURSOR 22,1
| | | PRINT "# of Azimuthal increments - -:";TAB(31);inct;TAB(40);inct;
| | | | TAB(51);"inct={#}"
| | CASE 23
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "Input Wave {PL,GAUSS} - - - -:";TAB(41);input_wave$;
| | | | TAB(51);"input_wave$={PL,GAUSS}"
| | | SET CURSOR rrow,1
| | | SET COLOR "black/white"
| | | PRINT "Input Wave {PL,GAUSS} - - - -:";TAB(41);input_wave$
| | | SET COLOR "white/black"
| | | IF input_wave$="PLANE" then inputwave$="PL" ELSE inputwave$=
| | | | "GAUSS"
| | | inputwave$=UCASE$(input_string$(rrow,32,"PL","GAUSS",
| | | | "PL","PL","PL","Enter either PL or GAUSS please",inputwave$))
| | | IF inputwave$="PL" then input_wave$="PLANE" ELSE input_wave$=
| | | | "GAUSSIAN"
| | | SET CURSOR 23,1
| | | PRINT erase_line$
| | | SET CURSOR 23,1
| | | PRINT "Input Wave {PL,GAUSS} - - - -:";TAB(32);inputwave$;
| | | | TAB(41);input_wave$;TAB(51);"input_wave$={PL,GAUSS}"
| | CASE IS >=24
| | | rrow=24
| | | |
| | | IF m1$="PLANE" and m2$="PLANE" then
| | | | resonator$="PLANE-PARALLEL"
| | | | confined$="PLANE-PARALLEL"
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "<<<<<<<  CONFINED BEAM  >>>>>>> ";resonator$
| | | | |
| | | ELSEIF m1$="PLANE" then
| | | | i=(1-2*cavity_length/radius2)^2
| | | | IF i < 0 OR i > 1 then
| | | | | resonator$=""
```

```
| | | | | confined$="N"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< UNCONFINED BEAM >>>>>>>";
| | | | ELSEIF cavity_length=radius2 then
| | | | | resonator$="HALF-CONCENTRIC"
| | | | | confined$="HALF-CONCENTRIC"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
| | | | ELSEIF 2*cavity_length=radius2 then
| | | | | resonator$="HALF-CONFOCAL"
| | | | | confined$="Y"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
| | | | ELSE
| | | | | resonator$=""
| | | | | confined$="Y"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>>";
| | | | END IF
| | | | |
| | | ELSEIF m2$="PLANE" then
| | | | i=(1-2*cavity_length/radius1)^2
| | | | IF i < 0 OR i > 1 then
| | | | | resonator$=""
| | | | | confined$="N"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< UNCONFINED BEAM >>>>>>>";
| | | | ELSEIF cavity_length=radius1 then
| | | | | resonator$="HALF-CONCENTRIC"
| | | | | confined$="HALF-CONCENTRIC"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
| | | | ELSEIF 2*cavity_length=radius1 then
| | | | | resonator$="HALF-CONFOCAL"
| | | | | confined$="Y"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
```

```
| | | | ELSE
| | | | | resonator$=""
| | | | | confined$="Y"
| | | | | SET CURSOR 2,1
| | | | | SET COLOR "black/white"
| | | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>>";
| | | | END IF
| | | | |
| | | ELSEIF (cavity_length=radius1) and (cavity_length=radius2) then
| | | | resonator$="CONFOCAL"
| | | | confined$="Y"
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
| | | | |
| | | ELSEIF (cavity_length/2=radius1) and (cavity_length/2=radius2) then
| | | | resonator$="CONCENTRIC"
| | | | confined$="CONCENTRIC"
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>> ";resonator$
| | | | |
| | | ELSEIF (1-cavity_length/radius1)*(1-cavity_length/radius2) < 0 &
| | | | &OR (1-cavity_length/radius1)*(1-cavity_length/radius2) > 1 then
| | | | resonator$=""
| | | | confined$="N"
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "<<<<<<< UNCONFINED BEAM >>>>>>>";
| | | | |
| | | ELSE
| | | | resonator$=""
| | | | confined$="Y"
| | | | SET CURSOR 2,1
| | | | SET COLOR "black/white"
| | | | PRINT "<<<<<<< CONFINED BEAM >>>>>>>";
| | | END IF
| | | |
| | | SET COLOR "white/black"
| | | SET CURSOR rrow,1
| | | PRINT erase_line$
| | | SET CURSOR rrow,1
| | | PRINT "READY? - - - - - - - - - - -:";TAB(51);"b$={Y/N}"
| | | SET CURSOR rrow,1
```

```
| | | SET COLOR "black/white"
| | | PRINT "READY? - - - - - :";TAB(41);"N"
| | | SET COLOR "white/black"
| | | b$=UCASE$(input_string$(rrow,32,"Y","N","y","n","n",
| | | | "Y/N ENTRY PLS","N"))
| | | SET CURSOR 24,1
| | | PRINT erase_line$
| | | SET CURSOR 24,1
| | | PRINT "* READY? - - - - - :";TAB(32);b$;TAB(41);b$;
| | | | TAB(51);"b$={Y/N}"
| | | IF b$="Y" then
| | | | EXIT DO
| | | ELSE
| | | | SET CURSOR 2,1
| | | | PRINT erase_line$
| | | END IF
| | END SELECT
| LOOP
| SET CURSOR 2,1
| SET COLOR "black/white"
| PRINT "<<<<<<<  ONE MOMENT PLEASE  >>>>>>>";
| SET COLOR "white/black"
| SET MODE "vga"
END SUB
|
!************************************************************ !
!                SUBROUTINE INPUT_:Gets a line of input.      !
!************************************************************ !
|
SUB input_
| LINE INPUT PROMPT "":a$
| a$=UCASE$(a$)
END SUB
|
!************************************************************ !
!     SUBROUTINE ZERO_:Display a non-valid variable input (zero).   !
!************************************************************ !
|
SUB zero
| SET CURSOR 2,1
| SET COLOR "black/white"
| PRINT "Zero is a non-valid variable input, Cr to continue:";
| CALL input_
| SET CURSOR 2,1
```

```
| SET COLOR "white/black"
| PRINT erase_line$
END SUB
|
!************************************************************!
!    SUBROUTINE SIMPSONS_RULE1: Perform a single integration.    !
!************************************************************!
|
SUB Simpsons_Rule1 (arr(),dn,inc, gral)                    !!!
| gral1,gral2=0                                             !
| FOR iv=2 to inc-1 STEP 2                                  !  Simpson's Rule.
| | gral1=gral1+arr(iv)                                     ! "inc" must be ODD.
| NEXT iv                                                   ! The interval must be
| FOR iv=3 to inc-2 STEP 2                                  ! divided into an even
| | gral2=gral2+arr(iv)                                     ! number of dn's.
| NEXT iv                                                   !
| gral=(arr(1)+arr(inc)+4*gral1+2*gral2)*dn/3              !!!
END SUB
|
!************************************************************!
!    SUBROUTINE SIMPSONS_RULE2: Perform a double integration.    !
!************************************************************!
|
SUB Simpsons_Rule2 (re(),im(),dn,inc,real,imag)            !!!
| real1,real2,imag1,imag2=0                                 !
| FOR iv=2 to inc-1 STEP 2                                  !
| | real1=real1+re(iv)                                      !  Simpson's Rule.
| | imag1=imag1+im(iv)                                      ! "inc" must be ODD.
| NEXT iv                                                   ! The interval must be
| FOR iv=3 to inc-2 STEP 2                                  ! divided into an even
| | real2=real2+re(iv                                       ! number of dn's.
| | imag2=imag2+im(iv)                                      !
| NEXT iv                                                   !
| real=(re(1)+re(inc)+4*real1+2*real2)*dn/3                 !
| imag=(im(1)+im(inc)+4*imag1+2*imag2)*dn/3                !!!
END SUB
|
END
```

APPENDIX B

CROSS-REFERENCE MAP
FOR COMPUTER PROGRAM "RESONATE VERSION 1.0"

a	405	406	409	413	417	418	435	453	461
	462	463	468	472	473	474	484	518	519
	520	523	537	538	539	544	548	549	550
	553	554	558	584	585	588	592	596	597
	614	632	640	641	642	647	651	652	653
	663	697	698	699	702	716	717	718	723
	727	728	729	732	733	737			
a$	55	55	56	112	112	113	113	114	114
	114	114	114	115	2761	2762	2762		
a1$	85	114							
a2$	85	114							
a3$	85	114							
a4$	85	114							
a5$	85	114							
alternate$	210	271	271	586	586	589	589	598	598
	615	615	1003						
amp	1122	1123	1124	1311	1312	1313	1314	1324	1325
	1326	1327	1336	1337	1338	1339			
arr	2783	2786	2789	2791	2791				
auto_scale$	216	273	273	1597	1617	1637	1657		
b$	2736	2740	2740	2741					
bot_top	387	389	400	400	566	568	579	579	
cav_length	766	767	767	769	778	779	779	781	792
	793	793	795	803	804	804	806	876	877
	877	879	888	889	889	891	902	903	903
	905	913	914	914	916				
cav_length	1103	1109	1110	1111	1111	1113	1116		
cavity_length	188	766	767	768	768	777	777	778	779
	780	780	783	783	791	791	792	793	794
	794	796	796	802	802	803	804	805	805
	807	807	822	876	877	878	878	887	887
	888	889	890	890	893	893	901	901	902
	903	904	904	906	906	912	912	913	914
	915	915	917	917	1034	1041	1055	1086	1087
	1103	1105	1105	1107	1107	1195	1224	2154	2521
	2524	2526	2531	2536	2536	2642	2649	2655	2670
	2677	2683	2697	2697	2704	2704	2711	2711	2712
	2712								
cavity_length_temp	822	823	823	824	825	826			
col	24	25	30	36	42	48	85	86	
color	1142	1187	1187	1188	1188	1388	1408	1414	1434
confined$	763	873	2636	2645	2651	2657	2663	2673	2679
	2685	2691	2699	2706	2714	2721			

cos_kr	1212	1213	1214	1241	1242	1243	1269	1270	1271
	1295	1296	1297						
cos_t	15	263	382	1208	1237	1265	1291		
daf	487	488	494	500	503	504	507	510	511
	666	667	673	679	682	683	686	689	690
date$	1091	1091	1091	1383	1383	1383			
default	24	29	31	35	37	41	43	47	49
	67	76							
default$	85	90	91	95	96	100	101	105	106
	125	135							
depth1	15	260	308	322	465	476	476	479	479
	480	480							
depth2	15	260	360	374	644	655	655	658	658
	659	659							
dev	215	1355	1355	1355					
dev_amp	1348	1352	1354	1355					
diameter$	2089	2090	2091	2092					
dn	2783	2791	2798	2808	2809				
dr1	284	326	336	404	416	460	471	487	490
	498	517	536	547	816	817	837	838	1113
	1113	1116	1116	1132	1135	1201	1218	1222	1275
	1416	1424	1427	1430	1431	1449	1488	1562	1565
dr2	284	336	378	583	595	639	650	666	669
	677	696	715	726	926	927	935	936	1193
	1230	1247	1301	1390	1398	1401	1404	1405	1504
	1543	1552	1555	1571	1574				
dt	380	382	1216	1245	1273	1299			
erase_line$	66	75	124	134	217	2165	2200	2205	2223
	2229	2234	2243	2248	2263	2267	2272	2281	2286
	2310	2327	2332	2367	2372	2390	2396	2401	2410
	2415	2430	2434	2439	2448	2453	2477	2494	2499
	2514	2519	2534	2539	2554	2559	2577	2583	2588
	2606	2612	2617	2628	2729	2738	2745	2776	
eta	1110	1113	1116						
etch_depth	277	329	407	410	586	589			
etch_depth$	278	330	1057						
etch_depth1	300	308	322						
etch_depth1$	301	1070							
etch_depth2	352	360	374						
etch_depth2$	353	1073							
EXTEXT$	71	130							
EXTYPE	59	71	130						
flood_cent	394	398	561	573	577	740			
fzt_focus	281	333	779	779	780	889	889	890	

gim	11	261	1214	1216	1243	1245	1271	1273	1297
	1299	11	261						
gim2	1216	1218	1245	1247	1273	1275	1299	1301	
gral	2783	2791							
gral1	2784	2786	2786	2791					
gral2	2784	2789	2789	2791					
gre	11	261	1213	1216	1242	1245	1270	1273	1296
	1299								
gre2	11	261	1216	1218	1245	1247	1273	1275	1299
	1301								
grid_graph1$	850	1444							
grid_graph2$	949	1499							
h	1210	1213	1214	1239	1242	1243	1267	1270	1271
	1293	1296	1297						
i	154	155	156	279	280	281	282	286	287
	288	290	293	294	296	297	297	297	298
	299	306	308	308	322	322	324	331	332
	333	334	338	339	340	342	345	346	348
	349	349	349	350	351	358	360	360	374
	374	376	381	382	382	383	403	404	407
	407	410	410	414	415	416	420	421	422
	422	423	424	427	428	429	429	430	431
	437	438	439	439	440	441	444	445	446
	446	447	448	454	459	460	465	469	470
	471	476	477	478	478	479	480	485	489
	490	492	494	496	497	498	500	501	502
	502	503	504	507	508	509	509	510	511
	514	516	517	520	524	527	528	529	530
	535	536	545	546	547	550	551	552	552
	553	554	559	582	583	586	586	589	589
	593	594	595	599	600	601	601	602	603
	606	607	608	608	609	610	616	617	618
	618	619	620	623	624	625	625	626	627
	633	638	639	644	648	649	650	655	656
	657	657	658	659	664	668	669	671	673
	675	676	677	679	680	681	681	682	683
	686	687	688	688	689	690	693	695	696
	699	703	706	707	708	709	714	715	724
	725	726	729	730	731	731	732	733	738
	746	748	748	750	752	754	754	756	815
	816	816	817	817	818	836	837	837	838
	838	839	856	858	858	860	862	864	864
	866	925	926	926	927	927	928	934	935
	935	936	936	937	950	962	1112	1113	1113

	1113	1114	1115	1116	1116	1116	1117	1121	1122
	1122	1123	1125	1131	1132	1133	1133	1133	1134
	1310	1311	1311	1312	1314	1315	1323	1324	1324
	1325	1327	1328	1335	1336	1336	1337	1339	1340
	1389	1390	1390	1390	1391	1403	1404	1404	1405
	1405	1406	1415	1416	1416	1416	1417	1429	1430
	1430	1431	1431	1432	1448	1449	1449	1449	1450
	1467	1468	1469	1469	1471	1472	1472	1474	1474
	1477	1478	1488	1489	1503	1504	1504	1504	1505
	1522	1523	1524	1524	1526	1527	1527	1529	1529
	1532	1533	1543	1544	1551	1552	1553	1553	1553
	1554	1561	1562	1563	1563	1563	1564	1570	1571
	1572	1572	1572	1573	1598	1599	1602	1603	1604
	1606	1618	1619	1622	1623	1624	1626	1638	1639
	1642	1643	1644	1646	1658	1659	1662	1663	1664
	1666	1689	1690	1690	1691	1693	1695	1695	1697
	1697	1698	1707	1708	1708	1709	1713	1714	1714
	1715	1719	1720	1720	1721	1757	1758	1758	1759
	1761	1763	1763	1765	1765	1766	1775	1776	1776
	1777	1780	1781	1781	1782	1785	1786	1786	1787
	1844	1845	1845	1846	1853	1854	1855	1856	1864
	1865	1865	1866	1906	1907	1907	1908	1915	1916
	1917	1918	1926	1927	1927	1928	1954	1955	1956
	1980	1981	1981	1982	1984	1986	1986	1988	1988
	1989	2001	2002	2002	2003	2006	2007	2007	2008
	2011	2012	2012	2013	2031	2032	2032	2033	2039
	2040	2041	2042	2050	2051	2051	2052	2060	2061
	2061	2062	2068	2069	2070	2071	2079	2080	2080
	2081	2642	2643	2643	2670	2671	2671		
ii	539	540	541	718	719	720	1466	1467	1467
	1491	1521	1522	1522	1546				
iii	745	747	747	747	748	751	753	753	753
	754	855	857	857	857	858	861	863	863
	863	864							
im	2798	2802	2806	2809	2809				
im1	14	262	1150	1206	1235	1263	1289		
im1_n1	1206	1213	1214	1263	1270	1271			
im1_n2	1235	1242	1243	1289	1296	1297			
im2	14	262	1101	1116	1122	1133	1150	1218	1247
	1275	1301	1311	1324	1336	1390	1416	1449	1458
	1460	1461	1463	1469	1471	1472	1474	1504	1513
	1515	1516	1518	1524	1526	1527	1529	1553	1563
	1572								
imag	2798	2809							

imag1	2799	2802	2802	2809					
imag2	2799	2806	2806	2809					
inc	2783	2785	2788	2791	2798	2800	2804	2808	2809
incr	191	254	254	255	255	255	255	256	256
	257	257	257	257	260	261	261	262	262
	262	262	263	263	263	274	284	326	336
	378	387	389	403	415	415	415	421	428
	438	445	457	457	457	457	457	459	470
	470	470	477	487	489	497	497	497	500
	501	503	504	508	516	527	531	535	535
	546	546	546	551	566	568	582	594	594
	594	600	607	617	624	636	636	636	636
	636	638	649	649	649	656	666	668	676
	676	676	679	680	682	683	687	695	706
	710	714	714	725	725	725	730	743	746
	746	751	753	758	758	788	813	815	835
	836	849	850	853	856	856	861	863	868
	868	898	923	925	933	934	948	949	986
	1112	1115	1121	1127	1128	1131	1135	1166	1181
	1192	1199	1218	1221	1226	1228	1247	1252	1257
	1275	1278	1281	1283	1301	1310	1317	1318	1323
	1330	1331	1335	1342	1343	1375	1389	1398	1401
	1403	1415	1424	1427	1429	1448	1466	1503	1521
	1551	1555	1561	1565	1570	1574	1950	2156	2561
	2564	2566	2580	2585	2585				
inct	192	261	261	263	380	381	986	1207	1216
	1236	1245	1264	1273	1290	1299	2157	2590	2593
	2595	2609	2614	2614					
init	178	232							
initial_int1_amp	1127	1128	1128	1129					
initial_int2_amp	1317	1318	1318	1319					
initial_power1	1135	1136	1136	1137	1138	1586	1591		
initial_power2	1555	1556	1556	1557	1558	1591			
input_	54	64	73	111	122	132	2220	2260	2308
	2387	2427	2475	2574	2603	2760	2773		
input_num	24	29	35	41	47	56	67	76	2212
	2241	2255	2279	2302	2318	2379	2408	2422	2446
	2469	2485	2506	2526	2546	2566	2595		
input_string$	85	90	95	100	105	115	125	135	2185
	2352	2625	2736						
input_wave$	193	268	268	1052	1099	2158	2619	2622	2624
	2626	2626	2630						
inputs	143								
inputs$	144	224	232						

inputwave$	2624	2624	2625	2625	2626	2630			
int1_amp	1330	1331	1331	1348	1348	1353	1884	1891	1939
int2_amp	1342	1343	1343	1824	1829	1877			
intvar_max	1834	1834	1835	1835	1836	1842	1842	1842	1845
	1848	1848	1848	1848	1850	1850	1854	1854	1856
	1857	1858	1859	1870	1870	1870	1870	1871	1871
	1878	1896	1896	1897	1897	1898	1904	1904	1904
	1907	1910	1910	1910	1910	1912	1912	1916	1916
	1918	1919	1920	1921	1932	1932	1932	1932	1933
	1933	1940	2030	2030	2030	2032	2036	2036	2040
	2040	2042	2043	2044	2045	2059	2059	2059	2061
	2065	2065	2069	2069	2071	2072	2073	2074	
intvar_max_old	1878	1940							
intvar_min	1831	1831	1832	1832	1836	1842	1842	1842	1845
	1848	1848	1850	1855	1855	1856	1856	1857	1858
	1858	1859	1870	1870	1871	1879	1893	1893	1894
	1894	1898	1904	1904	1904	1907	1910	1910	1912
	1917	1917	1918	1918	1919	1920	1920	1921	1932
	1932	1933	1941	2030	2030	2030	2032	2036	2041
	2041	2042	2042	2043	2044	2044	2045	2059	2059
	2059	2061	2065	2070	2070	2071	2071	2072	2073
	2073	2074							
intvar_min_old	1879	1941							
intvar1_max	1834	1884	1895	1895	1896				
intvar1_max_old	1886	1898	1940						
intvar1_min	1831	1884	1892	1892	1893				
intvar1_min_old	1885	1898	1941						
intvar1_per_pass	13	266	1884	1891	1892	1895	1913	1927	1934
	1936	1937	2037	2051					
intvar2_max	1824	1833	1833	1835	1897				
intvar2_max_old	1836	1878	1886						
intvar2_min	1824	1830	1830	1832	1894				
intvar2_min_old	1836	1879	1885						
intvar2_per_pass	13	266	1824	1829	1830	1833	1851	1865	1872
	1874	1875	2066	2080					
iv	2785	2786	2787	2788	2789	2790	2800	2801	2802
	2803	2804	2805	2806	2807				
j	280	281	281	288	290	293	293	332	333
	333	340	342	345	345	404	406	409	416
	418	435	460	462	471	473	490	492	494
	498	500	503	504	507	510	511	517	519
	520	536	538	539	547	549	550	553	554
	583	585	588	595	597	614	639	641	650
	652	669	671	673	677	679	682	683	686

	689	690	696	698	699	715	717	718	726
	728	729	732	733	747	748	748	749	753
	754	754	755	787	788	788	812	813	813
	834	835	835	857	858	858	859	863	864
	864	865	897	898	898	922	923	923	932
	933	933	1132	1133	1397	1398	1398	1400	1401
	1401	1423	1424	1424	1426	1427	1427	1477	1479
	1480	1481	1481	1483	1483	1487	1488	1532	1534
	1535	1536	1536	1538	1538	1542	1543	1552	1553
	1562	1563	1571	1572					
jj	1458	1461	1463	1477	1513	1516	1518	1532	
jjj	1469	1472	1474	1477	1524	1527	1529	1532	
k	275	1113	1116	1170	1185	1209	1238	1266	1292
key	26	27	27	33	39	45	53	55	87
	88	88	93	98	103	110	112	1953	2095
kr	1209	1211	1212	1238	1240	1241	1266	1268	1269
	1292	1294	1295						
labels	964	1685	1753	1840	1902	1973			
labels	2104								
lambda	189	200	200	207	207	275	277	281	281
	293	293	297	297	300	329	333	333	345
	345	349	349	352	520	539	550	553	554
	699	718	729	732	733	767	779	793	804
	828	877	889	903	914	1034	1041	1049	1055
	1204	1233	2153	2298	2298	2314	2314	2465	2465
	2481	2481	2501	2504	2506	2511	2516	2516	
last_dev_amp	1347	1354	1355						
last_int1_amp	1129	1884	1891	1939					
last_int2_amp	1319	1824	1829	1877					
last_j	1479	1480	1487	1534	1535	1542			
last_last_dev_amp	1347	1355							
last_power	1138	1592	1615	1616	1635	1636	1655	1656	
last_power1	1137	1613	1634	1654					
last_power2	1557	1594	1614	1633	1653				
left	392	396	400	571	575	579			
loss_max	1596	1677	1677	1678	1678	1681	1687	1687	1690
	1693	1700	1700	1702	1703	1705	1725	1725	1725
	1725	1726	1726	1749	1755	1755	1758	1761	1768
	1768	1770	1771	1773	1791	1791	1791	1791	1792
	1792	1819	1979	1979	1979	1981	1984	1991	1991
	1993	1994	1996						
loss_max_old	1596	1681	1749	1819					
loss_max1	1596	1642	1649	1662	1669	1677			
loss_max2	1602	1609	1622	1629	1678				

loss_min	1595	1675	1675	1676	1676	1681	1687	1687	1690
	1693	1700	1702	1703	1704	1705	1725	1725	1726
	1749	1755	1755	1758	1761	1768	1770	1771	1772
	1773	1791	1791	1792	1818	1979	1979	1979	1981
	1984	1991	1993	1994	1995	1996			
loss_min_old	1595	1681	1749	1818					
loss_min1	1595	1643	1648	1663	1668	1675			
loss_min2	1603	1608	1623	1628	1676				
loss_per_pass	12	265	1591	1593	1615	1635	1655	1708	1728
	1729	1746	1776	1794	1795	1816	2002	2024	
loss1_per_pass	12	264	1633	1638	1639	1653	1658	1659	1714
	1781	1798	1800	1801	1811	2007	2019		
loss2_per_pass	12	264	1593	1598	1599	1613	1618	1619	1720
	1731	1732	1741	1786	2012	2022			
m_1$	2174	2176	2178	2180	2182	2185	2185	2186	2202
	2296	2296							
m_2$	2341	2343	2345	2347	2349	2352	2352	2353	2369
	2463	2463							
m1$	181	269	269	276	285	285	402	455	458
	486	486	491	499	515	764	772	776	801
	874	882	885	900	989	990	992	1001	1008
	1008	1009	1054	1065	1068	1077	1079	1104	1396
	1422	2139	2167	2170	2172	2188	2190	2192	2194
	2196	2202	2216	2216	2634	2641			
m2$	182	270	270	328	337	337	581	634	637
	665	665	670	678	694	764	772	775	790
	874	882	886	911	996	997	999	1001	1020
	1020	1021	1065	1071	1077	1081	1107	2146	2334
	2337	2339	2355	2357	2359	2361	2363	2369	2383
	2383	2634	2669						
main	145	225	231						
max_amp	1120	1124	1124	1126	1309	1313	1313	1314	1316
	1322	1326	1326	1327	1329	1334	1338	1338	1339
	1341	1390	1416	1449	1504				
max_amp_i	1314	1327	1339	1457	1458	1458	1460	1461	1461
	1463	1463	1477	1512	1513	1513	1515	1516	1516
	1518	1518	1532						
max_radius1	200	295	299	326	386	389	565	568	812
	834	961	1034	1426	1447	1455	2089	2144	2288
	2298	2300	2302	2314	2316	2318	2323	2329	2329
max_radius2	207	347	351	378	386	389	565	568	922
	932	961	1041	1400	1502	1510	2091	2151	2455
	2465	2467	2469	2481	2483	2485	2490	2496	2496

max_rt	190	253	264	264	266	266	1164	1179	1842
	1844	1844	1848	1850	1853	1858	1858	1861	1870
	1870	1871	1904	1906	1906	1910	1912	1915	1920
	1920	1923	1932	1932	1933	2030	2031	2031	2036
	2039	2044	2044	2047	2059	2060	2060	2065	2068
	2073	2073	2076	2106	2110	2110	2111	2111	2120
	2124	2124	2125	2125	2155	2541	2544	2546	2551
	2556	2556							
max_transits	253	265	1148	1687	1689	1689	1695	1697	1700
	1703	1703	1705	1725	1726	1755	1757	1757	1763
	1765	1768	1771	1771	1773	1791	1792	1979	1980
	1980	1986	1988	1991	1994	1994	1998	2113	2117
	2117	2118	2118						
maxradius1	2302								
maxradius2	2469								
mirror	1151	1157	1171	1251	1277	1321	1333	1346	1438
	1493	1560	1569	1612	1652	1680	1748	1826	1888
n	213								
n_number	1055	1060	1061	1063					
n_number1	1034	1035	1036	1038					
n_number2	1041	1042	1043	1045					
n1	1199	1200	1200	1201	1202	1203	1204	1205	1206
	1216	1216	1217	1221	1222	1224	1226	1231	1232
	1247	1247	1248	1257	1258	1259	1260	1261	1262
	1263	1273	1273	1274	1278	1281	1284	1285	1301
	1301	1302							
n2	1192	1193	1195	1197	1202	1203	1218	1218	1219
	1228	1229	1229	1230	1231	1232	1233	1234	1235
	1245	1245	1246	1252	1255	1258	1259	1275	1275
	1276	1283	1284	1285	1286	1287	1288	1289	1299
	1299	1300							
open_viewports	152	179	234						
outer_tiers1	186	196	198	242	244	2143	2274	2277	2279
	2279	2283	2283	2292	2294				
outer_tiers2	187	203	205	249	251	2150	2441	2444	2446
	2446	2450	2450	2459	2461				
p2	488	494	507	510	511	667	673	686	689
	690								
past_int1_amp	1348	1353							
phase	1143	1170	1185						
phase_delay	15	260	1170	1185	1477	1477	1532	1532	
pi	275	380	767	779	793	804	828	877	889
	903	914	1136	1455	1455	1461	1463	1472	1474

	1479	1481	1483	1510	1510	1516	1518	1527	1529
	1534	1536	1538	1556	1566	1575			
power	1558	1567	1576	1586	1592	1615	1616	1635	1636
	1655	1656							
power1	1565	1566	1566	1567	1633	1634	1653	1654	
power2	1558	1574	1575	1575	1576	1594	1613	1614	
q	463	474	642	653					
q1	463	464	464	465	474	475	475	476	479
	480	642	643	643	644	653	654	654	655
	658	659							
r	1208	1209	1210	1210	1237	1238	1239	1239	1265
	1266	1267	1267	1291	1292	1293	1293		
r1	1201	1202	1202	1203	1204	1222	1223	1223	1232
r1_lambda	1204	1210	1261	1267					
r1_lambda_1	10	256	1204	1261					
r1r1	1223	1231							
r1r22	1203	1208	1232	1237	1259	1265	1285	1291	
r1r22_1	9	257	1203	1259					
r1r22_2	9	257	1232	1285					
r2	1193	1194	1194	1203	1230	1231	1231	1232	1233
r2_lambda	1233	1239	1287	1293					
r2_lambda_2	10	256	1233	1287					
r2r2	1194	1202							
rad1	16	258	297	299	519	538	549		
rad2	16	258	349	351	698	717	728		
radius1	183	200	293	297	391	488	488	492	492
	492	500	500	503	503	504	504	520	520
	539	539	550	550	553	553	554	554	764
	765	767	767	768	793	793	794	823	825
	826	829	829	874	875	877	877	878	914
	914	915	1011	1011	1012	1013	1013	1014	1016
	1105	2140	2207	2210	2212	2226	2231	2231	2298
	2314	2670	2677	2683	2697	2704	2711	2712	
radius2	183	207	345	349	570	667	667	671	671
	671	679	679	682	682	683	683	699	699
	718	718	729	729	732	732	733	733	764
	765	804	804	805	824	825	826	830	830
	874	875	903	903	904	1023	1023	1024	1025
	1025	1026	1028	1107	2147	2374	2377	2379	2393
	2398	2398	2465	2481	2642	2649	2655	2697	2704
	2711	2712							
re	2798	2801	2805	2808	2808				
re1	14	262	1149	1205	1234	1262	1288		
re1_n1	1205	1213	1214	1262	1270	1271			

re1_n2	1234	1242	1243	1288	1296	1297			
re2	14	262	1100	1113	1122	1133	1149	1218	1247
	1275	1301	1311	1324	1336	1390	1416	1449	1457
	1458	1461	1463	1468	1469	1472	1474	1504	1512
	1513	1516	1518	1523	1524	1527	1529	1553	1563
	1572								
real	2798	2808							
real1	2799	2801	2801	2808					
real2	2799	2805	2805	2808					
resonator$	2094	2635	2639	2644	2650	2654	2656	2660	2662
	2672	2678	2682	2684	2688	2690	2698	2702	2705
	2709	2713	2720						
response$	85	120							
right	393	397	400	572	576	579			
rings1	16	259	293	295	462	473			
rings2	16	259	345	347	641	652			
rnd	310	362							
row	24	25	30	36	42	48	85	86	
rrow	28	28	28	28	28	28	34	34	34
	34	34	40	46	57	57	57	57	57
	57	89	89	94	94	94	94	94	99
	104	116	116	2159	2161	2163	2164	2166	2168
	2185	2204	2206	2208	2212	2215	2224	2233	2235
	2237	2241	2247	2249	2251	2255	2264	2271	2273
	2275	2279	2285	2287	2289	2302	2311	2318	2321
	2331	2333	2335	2352	2371	2373	2375	2379	2382
	2391	2400	2402	2404	2408	2414	2416	2418	2422
	2431	2438	2440	2442	2446	2452	2454	2456	2469
	2478	2485	2488	2498	2500	2502	2506	2509	2518
	2520	2522	2526	2529	2538	2540	2542	2546	2549
	2558	2560	2562	2566	2569	2578	2587	2589	2591
	2595	2598	2607	2616	2618	2620	2625	2632	2728
	2730	2732	2736						
rt	1144	1164	1172	1172	1179	1613	1618	1619	1653
	1658	1659	1713	1719	1731	1731	1732	1732	1741
	1780	1785	1798	1798	1800	1800	1801	1801	1811
	1829	1830	1833	1851	1864	1872	1874	1874	1875
	1875	1891	1892	1895	1913	1926	1934	1936	1936
	1937	1937	2006	2011	2019	2022	2037	2050	2066
	2079								
Rz	1111	1113	1116						
s$	314	319	321	366	371	373			
scale	305	312	312	312	317	317	317	357	364
	364	364	369	369	369				

show$	214	1197	1226	1255	1281				
sim	312	313	313	314	317	318	318	319	322
	364	365	365	366	369	370	370	371	374
sim$	304	321	321	356	373	373	1095		
sim1	303	313	313	314	318	318	319	355	365
	365	366	370	370	371				
similate$	208	307	359	1095					
Simpsons_Rule1	1127	1135	1317	1330	1342	1555	1565	1574	2783
Simpsons_Rule2	1216	1218	1245	1247	1273	1275	1299	1301	2798
sin_kr	1211	1213	1214	1240	1242	1243	1268	1270	1271
	1294	1296	1297						
step_max_radius	283	284	335	336	787	897	1055		
step_max_radius1	1423								
step_max_radius2	1397								
step_rings	14	265	281	283	333	335	406	409	418
	435	585	588	597	614				
step_switch$	211	272	272	407	410	419	436	586	586
	589	589	598	598	615	615			
step_tiers	212	265	279	283	331	335	405	417	584
	596	1059							
t1	1207	1208	1213	1214	1215	1264	1265	1270	1271
	1272								
t2	1236	1237	1242	1243	1244	1290	1291	1296	1297
	1298								
temp	2212	2213	2216	2226	2318	2319	2323	2379	2380
	2383	2393	2485	2486	2490	2506	2507	2511	2526
	2527	2531	2546	2547	2551	2566	2567	2570	2580
	2595	2596	2599	2609					
tiers1	196	198	200	200	240	242	244	244	259
	286	287	295	461	472	1078	1080	2292	2294
	2297	2298	2298	2314	2314	2314			
tiers2	203	205	207	207	247	249	251	251	259
	338	339	347	640	651	1078	1082	2459	2461
	2464	2465	2465	2481	2481	2481			
time$	1091	1383							
tol	302	303	312	317	354	355	364	369	1095
tpz1	185	194	194	194	198	200	200	239	239
	239	240	244	260	293	293	300	301	306
	308	308	322	322	463	464	474	475	476
	479	480	990	1067	2142	2250	2253	2255	2255
	2256	2269	2269	2290	2290	2290	2294	2298	2298
	2314	2314							
tpz2	185	201	201	201	205	207	207	246	246
	246	247	251	260	345	345	352	353	358

	360	360	374	374	642	643	653	654	655
	658	659	997	1067	2149	2417	2420	2422	2422
	2423	2436	2436	2457	2457	2457	2461	2465	2465
	2481	2481							
transits	1148	1151	1187	1190	1191	1220	1308	1347	1355
	1386	1412	1550	1590	1615	1632	1635	1655	1707
	1711	1728	1728	1729	1729	1746	1775	1794	1794
	1795	1795	1797	1816	1823	1826	1883	1944	1948
	1948	2001	2024						
variable_change	238	2132							
w0	769	781	782	795	799	806	808	828	831
	832	879	891	892	905	909	916	918	1109
	1113	1116							
w1	770	782	785	787	797	799	808	810	812
	816	817	831	834	837	838	1419	1423	1426
	1430	1431	832	880	892	895	897	907	909
	918	920	922	926	927	932	935	936	1393
	1397	1400	1404	1405					
waist	767	769	779	781	785	793	795	797	804
	806	810	877	879	889	891	895	903	905
	907	914	916	920	1109	1113	1113	1116	1116
z	1200	1202	1202	1210	1229	1231	1231	1239	1260
	1267	1286	1293						
z_1	9	254	1200	1260					
z_2	9	254	1229	1286					
z_m1	829	831							
z_m2	830	832							
z0	768	769	780	781	794	795	805	806	827
	828	831	832	878	879	890	891	904	905
	915	916	1105	1107	1109	1110	1111		
z0_sqrd	823	824	824	825	825	826	826	827	829
	830								
z1	525	528	528	531	533	540	540	550	550
	553	553	554	554	712	719	719	729	729
	732	732	733	733	1224	1229			
z11	526	529	529	534	541	541	550	553	554
	713	720	720	729	732	733			
z2	704	707	707	710	1195	1200			
z22	705	708	708						
zero	2214	2320	2381	2487	2508	2528	2548	2568	2597
	2769								
zones1	184	195	198	240	241	258	296	518	537
	548	2141	2236	2239	2241	2241	2245	2245	2291
	2294								